Este libro que hoy me enorgullezco de prologar, describe estas tecnologías de una forma simple, amena y con una generosidad de gráficos y ejemplos que nos facilitan su rápida compresión. Estoy seguro de que cualquier lector, profesional o no en la materia, encontrará en esta obra una buena herramienta para resolver sus dudas o curiosidades relativas al funcionamiento de la telefonía IP. Entenderlo es un paso para adaptarnos a un futuro ya presente.

Alberto Mazagatos Uriarte
Director General de Cisco Systems España

Qué es la telefonía IP

Para contactar con el autor: *carballar@telefonica.net*

INTRODUCCIÓN

Se denomina telefonía a la comunicación hablada entre dos o más participantes distantes utilizando una red pública que facilita el intercambio de la voz. El aparato telefónico que hace posible esta comunicación se inventó a finales del siglo XXI, creándose poco después las redes públicas que permitieran su interconexión. Hoy en día, el servicio telefónico está extendido por todo el mundo, llegando a cubrir los lugares más remotos.

La tecnología Internet se creó a finales de los años sesenta del siglo XX, dando paso a la creación de una red de datos que poco a poco está alcanzando también una cobertura mundial. Aunque el servicio telefónico está mucho más extendido que Internet, en cifras globales, este último es mucho más utilizado que el primero. De hecho, a pesar de que el tráfico de las redes de telefonía no ha dejado de crecer en el mundo año tras año, el volumen de tráfico de Internet no sólo lo superó en el año 1999, sino que hoy en día supone varias veces su valor.

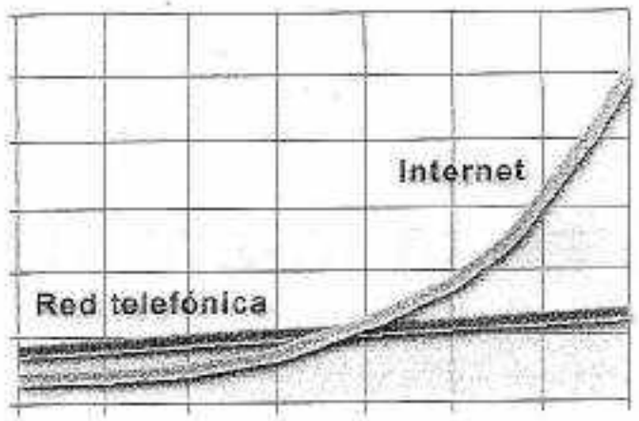

Figura 1.1. Evolución del tráfico de Internet y de red telefónica.

Telefonía IP, telefonía sobre Internet, voz sobre banda ancha (VoBB, *Voice over Broad Band*), voz sobre IP o VoIP (*voice over IP*) vienen a significar una misma cosa: un servicio que permite la transmisión de la voz utilizando la red Internet.

El término IP es el nombre de la tecnología principal que utiliza la red Internet: el protocolo IP (*Internet Protocol*, 'Protocolo de Internet'). Esta tecnología IP también la utilizan las redes locales de datos, como Ethernet o Wi Fi, por lo que la VoIP puede ser empleada también para facilitar las comunicaciones internas de una empresa, como sustituto del sistema tradicional de centralitas.

Aparte de lo anterior, los sistemas de VoIP pueden utilizarse también en aplicaciones particulares en las que se necesite el envío de audio en tiempo real. Este es el caso de las aplicaciones de televigilancia o teleasistencia, entre otras.

Por otro lado, la aparición de las tecnologías inalámbricas, como Wi-Fi, está aportándole a Internet la movilidad de la que antes carecía. De hecho, puede que no sepamos exactamente por dónde evolucionará la tan hablada convergencia de fijo y móvil, pero seguro que la tecnología IP juega un papel importante.

Lo cierto es que, desde que apareció Internet, esta tecnología está sustituyendo paulatinamente al resto de las tecnologías de comunicaciones. En un principio se fue introduciendo tímidamente como una tecnología de transmisión de datos, pero, dado su bajo coste y la gran facilidad que tiene para adaptarse a cualquier necesidad, poco a poco se está apoderando del ámbito del resto de redes de comunicaciones: voz, vídeo, datos, fijo y móvil. Sus propiedades de ahorro y versatilidad se van imponiendo a la aparente ventaja de confiabilidad ofrecida por las redes tradicionales.

EN QUÉ CONSISTE LA TELEFONÍA IP

Se llama señal analógica o información analógica a aquella que varía de forma continua. La voz humana, al variar su volumen de forma continua, es un ejemplo de señal analógica. Las redes de telefonía traducen la señal analógica de las ondas sonoras de la voz en señales eléctricas analógicas que son transmitidas tal cual hasta el otro extremo de la comunicación, donde son de nuevo convertidas en ondas sonoras.

Por otro lado, se llama señal digital a aquella que varía de forma discreta. Quiere esto decir que las señales digitales disponen de un número reducido de estados posibles y van cambiando de uno a otro dependiendo de la información que transmiten. Una señal o información es digital binaria cuando el número de estados posibles es dos, representados como 0 y 1 y conocidos como bits. La red Internet maneja señales digitales binarias.

Digitalización y transmisión

Por tanto, el primer reto de la telefonía IP es convertir la señal analógica que produce la voz en digital, de forma que pueda ser tratada por Internet. A este proceso se lo conoce como **digitalización de la voz**. El proceso de digitalización consiste en tomar una muestra de la voz, cuantificarla y convertir este valor en un número binario. Si, por ejemplo, cada muestra se representa con 8 bits, y se toman 8.000 muestras por segundo (una cada 0,125 milisegundos), la señal de la voz se podría convertir en un flujo de datos de 64.000 bits por segundo (8 x 8.000).

En realidad, las tecnologías modernas de digitalización de la señal analógica de la voz incluyen ciertas técnicas de **codificación** que hacen que el ancho de banda necesario (velocidad de transmisión) para la voz sea mucho menor que estos 64 Kbps. Esta disminución del ancho de banda necesario es importante para conseguir altos niveles de calidad incluso cuando el acceso o las condiciones de la red no son idóneos.

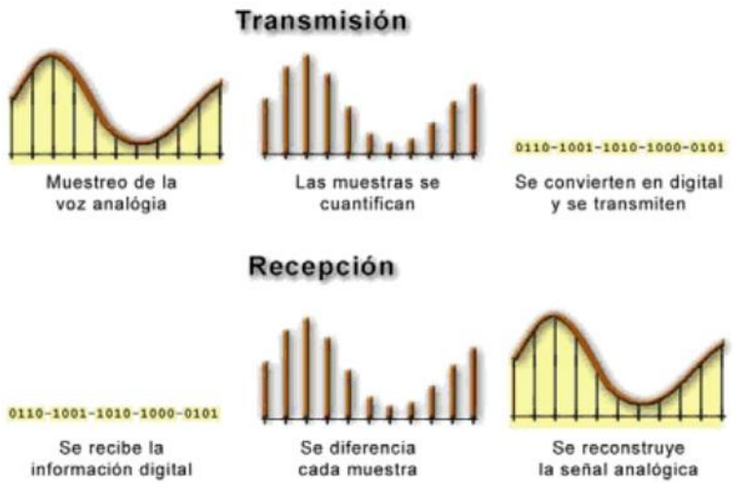

Figura 1.2. Digitalización de la voz.

Internet es una red que fue pensada para transmitir datos que no necesitaban llegar al destino de una forma ordenada y en tiempo real. Por ejemplo, no tiene importancia que la información que representa un mensaje de correo electrónico o el contenido de una página Web llegue de forma un poco desordenada; el programa de comunicación del terminal de destino pierde unos milisegundos en ordenarla y listo. El usuario que la recibe no percibe dicho desorden en absoluto.

El caso de la voz es distinto, las muestras de voz se codifican y decodifican de forma continua y en tiempo real, por lo que no hay tiempo para estar ordenando nada. Si algo no llega a tiempo, es mejor darlo por perdido. La pérdida de una muestra puede afectar levemente a la calidad de reproducción, pero no entorpecerá su entendimiento. Pues bien, en Internet ha habido que crear nuevos protocolos que puedan manejar esta forma distinta de transmitir la información. Se trata de los protocolos de **transmisión en tiempo real**.

Las redes telefónicas tradicionales ya hace tiempo que están digitalizadas. Esto quiere decir que tanto sus equipos de conmutación como de transmisión utilizan señales

digitales. La voz es digitalizada en las centrales antes de pasar por el primer equipo de conmutación. Las señales digitales tienen la gran ventaja de ser mucho más inmunes al ruido que las señales analógicas. La razón es que el ruido es una señal analógica, que si se mezcla con la señal de la voz, también analógica, resulta muy difícil de separar, llegando ambas al destino. En cambio, cuando el ruido se mezcla con una señal digital se pueden separar ambas señales y dejar limpia la señal digital original antes de entregarla en destino.

Las ventajas de la digitalización son tan evidentes que las redes telefónicas fueron evolucionando durante los años 80 y principios de los 90 hasta convertirse en redes digitales. La única parte analógica que queda en las redes telefónicas es el tramo que va desde el terminal telefónico a la central del operador. Este tramo se conoce como bucle de abonado o bucle local (*local loop* en inglés).

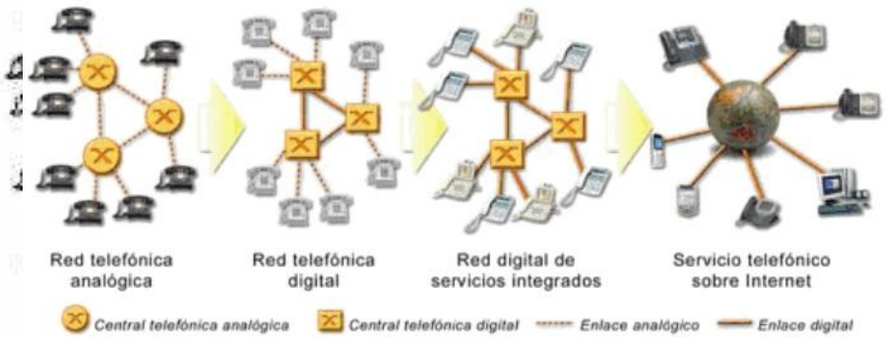

Figura 1.3. Evolución de la red telefónica.

En la década de los 90 se desarrolló un nuevo tipo de red, conocido como Red Digital de Servicios Integrados o RDSI (en inglés ISDN, *Integrated Services Digital Network*) que es completamente digital, de terminal a terminal. Aunque en esa época se pensaba que la RDSI sería la evolución natural de la red telefónica, lo cierto es que la aparición de la tecnología IP le ha quitado todo el protagonismo. De hecho, la utilidad de la RDSI ha quedado reducida hoy en día al establecimiento de videoconferencias (existen equipos comerciales específicos) y como acceso a Internet cuando no se dispone de banda ancha (DSL o cable). Una línea básica RDSI permite la transmisión de datos a 128 Kbps (2x64 Kbps).

Por tanto, la comunicación de la voz a través de una red IP consiste en los siguientes pasos:

1. Registrar las ondas sonoras de la voz con un micrófono.
2. Digitalización y codificación de la señal para convertirla en un flujo de bits.
3. Transmitir la información en tiempo real.
4. Decodificación de la información y conversión de digital a analógico.
5. Producir las ondas sonoras con un altavoz.

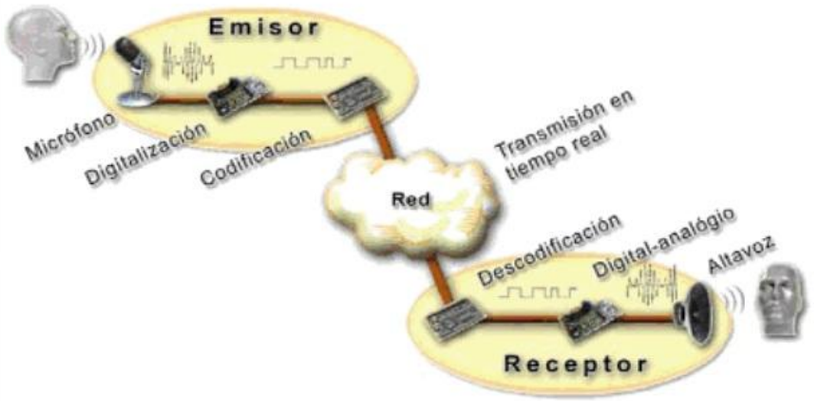

Figura 1.4. Transmisión de la voz en redes digitales.

Elementos necesarios

El proceso de la telefonía IP explicado anteriormente puede llevarse a cabo utilizando distintos tipos de equipos:

- **Un ordenador**. En este caso es necesario contar con un micrófono para poder captar la voz y unos altavoces para poder reproducirla. Además, es necesario contar con el software que realiza la digitalización y codificación (*softphone*). Para mejorar la comodidad en la comunicación existen terminales telefónicos que son conectables al ordenador a través de USB o de los conectores de micrófono y altavoz.
- **Un teléfono IP**. Estos equipos tienen una apariencia similar a la de un teléfono tradicional, pero incorporan los elementos necesarios para convertir la voz en información IP y viceversa. Un tipo particular de teléfonos IP son los terminales multitecnología. Estos equipos funcionan en redes de tecnología distinta, como pueden ser, redes de VoIP (Wi-Fi) y móvil (GSM).

- **Un teléfono tradicional con un adaptador IP**. En este caso, las labores de conversión a IP y viceversa las realiza el adaptador IP. Este equipo se conecta por un lado al teléfono tradicional y por otro a la red IP (al *router* DSL o cable). En el mercado existen desde pequeños adaptadores individuales hasta centralitas IP.

Figura 1.5. Distintos terminales telefónicos de VoIP.

La utilización de la red telefónica tradicional es muy sencilla: simplemente hay que introducir el número telefónico del destinatario o levantar el auricular cuando suena el timbre. Sin embargo, la utilización de un PC con un software de telefonía o la de un terminal IP avanzado puede resultar algo más elaborado. Para aquellos que desean disfrutar de las ventajas de la telefonía IP sin tener que molestarse en aprender algo nuevo existe la solución de utilizar teléfonos IP simples o seguir con los aparatos telefónicos tradicionales.

Interconexión de redes

En principio, la telefonía IP sólo establece comunicaciones entre terminales conectados a una red IP (Internet y cualquier otra red privada del tipo IP). No obstante, donde se encuentra la inmensa mayoría de los usuarios de telefonía no es en Internet, sino en las redes de telefonía fija y móvil (celular). Por tanto, tiene sentido hacer posible que los usuarios de telefonía de Internet puedan comunicarse con los de las redes telefónicas tradicionales y viceversa.

Dado que cada red utiliza internamente sus propios sistemas de funcionamiento, para que esta interconexión sea posible hace falta contar con un equipo que entienda ambos sistemas y adapte las comunicaciones de uno a otro. A este traductor de sistemas se lo conoce como *gateway* (pasarela). Este equipo se encarga de traducir las distintas informaciones de control utilizadas por ambas redes, además de convertir la información binaria IP en información analógica (o digital G.711). Dicho de otra forma, el *gateway* se encarga de realizar todas las traducciones necesarias para que la información fluya de una a otra red sin problemas.

La interconexión de redes supone que también se produce una interconexión entre operadores. En este caso, el operador de la red donde se origina la llamada debe pagarle al operador donde se termina la llamada por el uso de su red. Este coste lo repercute a sus usuarios en sus tarifas. La ventaja que tiene la VoIP es que puede viajar por Internet hasta la localidad del destinatario, donde la tarifa de terminación de la llamada es local, aunque la llamada se realice desde otro país. Por otro lado, si el destinatario de la llamada también está en Internet, no habría coste de terminación.

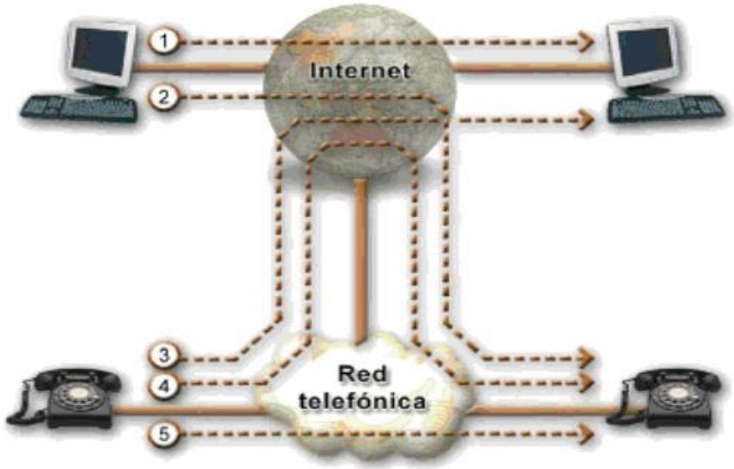

Figura 1.6. Interconexión de redes.

Proveedores de servicio

En un principio, los primeros que ofrecieron un servicio de VoIP fueron los desarrolladores de software al hacer posible una comunicación de voz entre dos ordenadores conectados a Internet utilizando un software específico. Posteriormente

aparecieron las pasarelas o *gateway* que permitían la interconexión de las redes IP y de telefonía, lo que permitió establecer comunicaciones de voz entre usuarios de la red Internet y de la red telefónica. Finalmente, con el objetivo de abaratar las llamadas de larga distancia, han aparecido intermediarios que permiten la realización de llamadas entre usuarios de la red telefónica, pero utilizando Internet para su transporte internacional.

Por tanto, existen tres tipos de proveedores de telefonía IP:

- Los que facilitan la comunicación de voz entre usuarios de Internet.
- Los que facilitan la comunicación de voz entre un usuario de Internet y otro de la red telefónica.
- Los que facilitan la interconexión de dos usuarios de la red telefónica a través de Internet.

Cada tipo de proveedor puede aplicar distintos modelos de negocio, no obstante, en el primer caso, el coste de la llamada suele ser gratuito, mientras que en los otros dos casos suele haber un coste, aunque sea bastante inferior al aplicado a una llamada puramente telefónica.

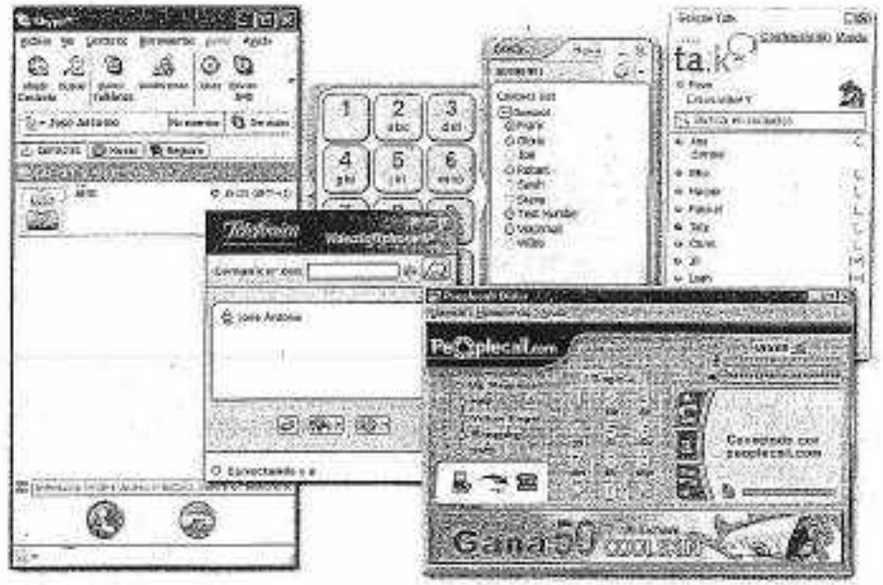

Figura 1.7. Ejemplos de softphones

Los operadores de telefonía tradicional, aprovechando que suelen ser también proveedores de acceso de banda ancha, están empezando a ofrecer servicios de VoIP, tanto como líneas telefónicas individuales como centralitas IP. Suele ser habitual también que este tipo de servicios se ofrezcan mediante una reducida tarifa fija mensual que incluye todas las llamadas telefónicas con destino a teléfonos fijos nacionales.

Entre los proveedores de servicios de telefonía IP tenemos a Skype, Peoplecall Voipbuster, Telefónica, Yahoo, Google o Microsoft (Messenger).

VENTAJAS DE LA TELEFONÍA IP

Cuando se empezó a introducir la tecnología de voz sobre IP a finales de los años 90, la finalidad principal era ahorrarse un dinero cuando se necesitaba hablar con un amigo del otro lado del país o del planeta. En vez de hacer una cara llamada telefónica se instalaba un programa en el ordenador, un micrófono y unos altavoces y se realizaba la comunicación al coste de una llamada local. La calidad de la voz era muy baja, pero suficiente para una llamada de tipo personal.

Hoy en día, la incorporación de la banda ancha ha hecho que la calidad de la voz en las comunicaciones IP haya aumentado considerablemente, por lo que esta tecnología empieza a verse como una seria sustituta de los sistemas tradicionales. No cabe duda de que IP es más económico, pero éste no es más que una de sus múltiples ventajas.

Evidentemente, cabría preguntarse que si la VoIP es tan buena, por qué se está tardando tanto en su implantación. La respuesta es múltiple: por un lado ha sido necesario un esfuerzo regulador importante y más complejo de lo esperado. Por otro, la necesidad de compatibilizar ambas redes (telefonía tradicional e IP) hace que su despliegue resulte también más caro. Introducir la telefonía IP en Internet es relativamente simple, mientras que hacer que estos nuevos teléfonos se entiendan con los antiguos requiere la participación de más actores, lo que complica el proceso.

Por cierto, además de la voz sobre IP existen tecnologías que permiten transmitir la voz sobre otro tipo de redes de datos. Por ejemplo, existe la voz sobre *frame relay* (VoFR) o sobre ATM (VoATM). No obstante, mientras que estas últimas tecnologías no están ampliamente extendidas, la tecnología IP está incluida en todos los ordenadores personales y es muy común encontrarla en nuevos dispositivos electrónicos como PDA.

Inferior coste de los equipos

Es curioso, pero, a pesar de que la tecnología de los sistemas de telefonía es muy antigua, las redes de telefonía utilizan sistemas que, en la mayoría de los casos, son propios de cada fabricante (propietario). Esto quiere decir que un equipo de conmutación de un suministrador utiliza un sistema operativo propio, con un software propio que funciona exclusivamente con su propio hardware. Cada suministrador diseña y produce absolutamente todo el equipamiento necesario: procesadores, memorias, cableado, fuentes de alimentación, sistemas operativos, aplicaciones y toda la cacharrería necesaria. A todo esto hay que sumarle el coste de la formación específica que requieren estos equipos para su gestión y mantenimiento. Por si fuera poco, todo lo anterior hace que el operador de la red tenga una alta dependencia del proveedor escogido.

El mundo IP es completamente diferente. Aunque existe algún hardware propietario, la mayoría del equipamiento necesario está completamente estandarizado, con lo que cualquier equipo puede ser reemplazado por otro de un fabricante distinto sin mucha dificultad. Esta alta estandarización fomenta la competencia, haciendo que los costes de desarrollo, producción e implantación sean mucho menores.

Otra de las grandes ventajas de la tecnología IP es que resulta tremendamente flexible, lo que hace posible que se pueda introducir una nueva funcionalidad en Internet en cuestión de meses, mientras que en el mundo de la telefonía tradicional se necesitan años (de 18 a 24 meses).

Por si todo lo anterior fuese poco, los precios del equipamiento de las redes IP bajan a un ritmo mucho mayor que el de las redes de telefonía tradicional. Esto nos lleva a la conclusión de que la diferencia de costes de ambas redes irá aumentando con el tiempo. Actualmente, el uso de la red IP es entre 10 y 20 veces más barato que el de las redes de telefonía.

Mejor utilización de los recursos

Las redes de telefonía tradicional utilizan una técnica, conocida como conmutación de circuitos, que hace que un circuito, un enlace, se dedique en exclusividad a una comunicación, se esté o no utilizando en cada momento. Quiere esto decir que, aunque ambos interlocutores estén callados, el circuito sigue dedicado a dicha comunicación. En cambio, en las redes IP se liberan los recursos cuando no hay información que transmitir.

Por otro lado, las redes de telefonía actuales utilizan medios digitales de transmisión, pero siguen la antigua recomendación G.711 de la UIT (Unión Internacional de Telecomunicaciones) que dedica 64 Kbps para cada circuito de voz. El sistema de codificación de la voz utilizado por las técnicas de VoIP es mucho más reciente,

necesitándose un ancho de banda que puede llegar a ser hasta la octava parte del anterior. Para ello se suprimen los silencios y se utilizan técnicas de compresión.

Aunque, desde el punto de vista del usuario, podría pensarse que la utilización de recursos de red es algo que afecta exclusivamente al operador, lo cierto es que, en general, se puede decir que cuanto menor sea su coste de operación, menor precio tendrán sus servicios.

La idea es que desde una misma red IP se puedan ofrecer servicios de distinta naturaleza (voz, datos, vídeo, etc.). No obstante, es importante que la red ofrezca en estos casos un tratamiento distinto a cada tipo de información. Por ejemplo, no tiene sentido que un paquete de voz esté esperando en un *router* a ser encaminado, mientras que una transferencia de archivos ocupa todo el ancho de banda. Es más lógico que si la red empieza a estar sobrecargada, se le dé prioridad al tráfico de voz, aunque sea a costa de retrasar el de datos. De hecho, si no se dispone de suficientes recursos en la red, es mejor que no se inicie la comunicación de voz, antes que establecerla con una pésima calidad.

Lo cierto es que los nuevos protocolos de red hacen posible una buena coexistencia de los servicios de datos y de aquellos otros que requieren una transmisión en tiempo real, por lo que estos posibles inconvenientes están tecnológicamente resueltos.

Servicios de valor añadido

El protocolo IP permite crear nuevos servicios o nuevas funcionalidades de una forma muy fácil. Al ser un protocolo pensado para la transmisión de datos, basta añadirle algunos datos a la transmisión de la voz para obtener un servicio nuevo. Por ejemplo, durante la conversación de voz se podrían intercambiar archivos (fotografías, gráficos, documentos, etc.), o bien, utilizar una pizarra virtual o compartir aplicaciones (de presentaciones, hoja de cálculo, etc.).

Por otro lado, los servicios de datos actuales pueden verse enriquecidos con la utilización de la telefonía. Por ejemplo, se podría hablar con un representante del servicio de atención al cliente de una empresa simplemente haciendo clic en un icono de su página Web. Incluso se podría direccionar la llamada al operador más apropiado dependiendo de la página en que se encuentre el icono o de la navegación previa que se haya realizado.

Un área interesante en la que la VoIP puede aportar un gran valor es en el trabajo en grupo y el teletrabajo. Por ejemplo, se puede establecer un grupo de trabajo en el que cada componente conoce en cada momento el estado en el que se encuentran los demás (hablando, ausente, etc.). En esta situación, si se desea hablar con alguien, simplemente

se espera a que esté disponible, evitándose perder el tiempo en hacer llamadas que tienen como respuesta una señal de línea ocupada o un mensaje de contestador.

Por último, en el entorno corporativo se podrían combinar distintos tipos de servicio. Por ejemplo, al recibir una llamada se podría ver automáticamente la ficha del compañero que nos llama para conocer exactamente con quién vamos a hablar, área en la que trabaja, posición en la jerarquía, su foto, etc. De hecho, nos podría informar hasta de los asuntos que tenemos pendientes con esa persona. Realmente, basta utilizar un poco la imaginación para ver sus posibilidades.

Mantenimiento y gestión integrada

Con la VoIP se puede llegar a disponer de toda la información de facturación y estadísticas en tiempo real y desde cualquier lugar. Si se desea conocer cómo va el consumo por cada área de la empresa, qué empleados hacen un uso más intensivo o qué terminal no se utiliza nunca, basta con conectarse al sistema de gestión.

Por otro lado, la integración de los servicios de voz y datos en una misma infraestructura aporta unos beneficios adicionales, ya que se puede emplear una misma base de datos de usuarios para gestionar todos los servicios de comunicaciones y realizar informes. Por ejemplo, si se da de baja un empleado, basta con hacer una gestión para darle de baja todos sus servicios de comunicaciones. Además, al utilizar la red de voz y de datos la misma tecnología IP, no será necesario contar con personal de mantenimiento diferenciado. Aunque la VoIP supone adquirir unos conocimientos particulares, el personal de mantenimiento con experiencia en IP no suele tener dificultades para ello. Esto, no sólo simplifica la formación, sino que hace que los recursos de gestión y mantenimiento sean más eficientes.

Por cierto, las redes locales de datos (redes corporativas) suelen estar utilizadas al 30% de su potencial. Quiere esto decir que, cuando se introducen los servicios de voz sobre IP, no suele ser necesario ampliar esta red. Por otro lado, la ampliación del ancho de banda del enlace con Internet suele ser menor que la capacidad equivalente de los circuitos de telefonía.

Por último, existen soluciones corporativas que disponen de herramientas que permiten identificar y atender cualquier incidente que pueda aparecer en la red, inclusive, antes de que sea percibido por los usuarios. Estas herramientas, además, pueden utilizarse de forma local o remota.

Movilidad

Otro de los servicios interesantes que ofrece la telefonía IP es la movilidad. A diferencia de la telefonía tradicional, donde la identificación del usuario está ligada a un punto físico concreto (en el caso de la telefonía fija) o a un terminal concreto (en el caso de la telefonía móvil), para la telefonía IP basta con disponer de un nombre de usuario y clave. Por ejemplo, en el caso de la telefonía móvil (celular), si te quedas sin batería te quedas sin servicio. Esto no ocurre con la telefonía IP, ya que si tu propio PDA o portátil se queda sin batería siempre se puede utilizar cualquier otro equipo, prestado o alquilado, para disponer del servicio.

Por otro lado, en el entorno de la empresa es habitual disponer de una red local inalámbrica (Wi-Fi), por lo que el empleado puede desplazarse por todas sus dependencias sin dejar de estar conectado. Además, a diferencia de la telefonía móvil, esta disponibilidad tiene muy bajo precio.

Esta característica de movilidad de la VoIP hace innecesaria la utilización de servicios como el desvío de llamadas, pudiéndose atender las llamadas desde la oficina por la mañana, en el hotel de la reunión al mediodía y desde casa por la tarde sin que el otro interlocutor note la diferencia. De hecho, se podrían utilizar hasta distintos dispositivos: un terminal de VoIP en el trabajo, la PDA en el hotel y nuestro portátil en casa.

Por último, en los últimos años se está produciendo una expansión considerable de las zonas públicas de acceso inalámbrico a Internet (*Hot spots*). En estas zonas se puede acceder a Internet de una forma fácil y rápida utilizando un terminal propio (PDA, portátil, etc.). Esta forma de acceso simplifica el uso de la telefonía IP en lugares públicos como hoteles, aeropuertos, palacios de congresos, etc. Por tanto, la movilidad se puede disfrutar tanto dentro como fuera de la empresa.

MITOS DE LA TELEFONÍA IP

Cuando aparece una nueva tecnología suele ser habitual que, en buena medida, se difunda gracias al boca a boca. Desgraciadamente, entre lo que una persona escucha y lo que finalmente cuenta, suele haber pequeñas diferencias que se van exagerando con el tiempo. La tecnología de voz sobre IP no se ha librado de este efecto, por lo que existen algunos mitos que se han ido creando en torno a ella. Algunos de estos son los siguientes:

- **La calidad no es buena**. Aunque las primeras comunicaciones de VoIP tenían una calidad deficiente, en la actualidad, los servicios de telefonía IP tienen una calidad comparable al de las redes de telefonía tradicionales.

- **Sólo funciona en Internet**. La tecnología de voz sobre IP funciona con el protocolo IP, utilizado tanto por la red pública Internet como por las redes privadas de área local. Esto permite crear redes privadas de VoIP.
- **Los terminales son caros**. Aunque en el mercado existen terminales específicos muy sofisticados (que incluyen videoconferencia, agenda, etc.) que pueden tener un precio elevado, la telefonía IP permite seguir utilizando los terminales analógicos convencionales, por lo que el precio de los mismos no es una barrera.
- **Estas comunicaciones pueden ser interceptadas**. Interceptar las comunicaciones de la telefonía tradicional es mucho más sencillo que hacerlo con la VoIP. En el primer caso, basta identificar la caja de registro, que suele estar en la fachada del edificio y pinchar un par de hilos conectados a un teléfono. Para tener la posibilidad de interceptar las comunicaciones IP, no sólo hay que tener unos conocimientos muy especializados, sino que requiere un trabajo de investigación nada sencillo. Además, mientras que en Internet queda todo registrado, en la red de telefonía tradicional no es así, lo que hace más sencillo identificar el culpable en el primer caso.
- **Las llamadas a números de emergencia no funcionan**. Los números de emergencia (112 en Europa, 911 en Norteamérica, etc.) se crearon para ser atendidos desde la red telefónica tradicional. Ciertamente, no todos los servicios de telefonía IP permiten llamar a los servicios de emergencia, entre otras cosas, porque la atención de estas llamadas suelen estar relacionadas con el lugar desde donde se originan, y la telefonía IP ofrece movilidad. No obstante, los servicios de VoIP ofrecidos por los operadores de telefonía sí suelen incluir el acceso a los servicios de emergencia. En cualquier caso, siempre se pueden realizar este tipo de llamadas desde otros teléfonos (móviles, cabinas, etc.).
- **Los servicios adicionales son caros**. En realidad, habría que hacer una comparación caso por caso; pero, en general, la telefonía IP ofrece la mayoría de los servicios de valor añadido de forma gratuita. Sólo se cobran aquellos que aportan un valor añadido importante por los que el cliente está dispuesto a pagar un sobreprecio.
- **Es complicado utilizar sus terminales**. La telefonía IP permite seguir utilizando los teléfonos analógicos tradicionales, por lo que su uso es idéntico. No obstante, para los usuarios que necesitan o aprecian los nuevos servicios de valor añadido, existen nuevos terminales cuya utilización puede resultar algo menos simple, pero, en cualquier caso, no más complicada que muchos de los terminales telefónicos actuales.

REGULACIÓN

Tradicionalmente, para que un operador pueda ofrecer un servicio de telefonía necesita solicitar una licencia al gobierno, por la que, no sólo debe pagar unos impuestos,

sino que se ve obligado a cumplir una serie de requisitos: ofrecer acceso universal, permitir la interconexión con otras redes, facilitar la portabilidad de números, etc. Por otro lado, para que un operador pueda ofrecer servicios de datos (por ejemplo, acceso a Internet) no necesita tantos requisitos ni tiene tantas limitaciones. La Unión Europea denomina a estos servicios como Servicio de Comunicaciones Electrónicas (SCE o ECS, *Electronic Communication Service*) y Servicio Telefónico Disponible al Público (STDP o PATS, *Publicly Available Telephone Service*).

La pregunta que viene a continuación es: ¿La telefonía IP es un servicio telefónico o es un servicio de datos? Dependiendo de esta consideración, los operadores que ofrecen este servicio deben someterse a una u otra regulación. El hecho es que el operador dominante de telefonía tradicional (Telefónica en el caso de España), los operadores de cable (en España se agrupan en torno a AOC), las empresas de telecomunicaciones (agrupadas en España en torno a la patronal Astel) o, incluso, los proveedores de servicios de VoIP tienen cada uno su propia opinión al respecto.

En cualquier caso, lo cierto es que todavía no existe una respuesta clara a esta pregunta. Los gobiernos están esperando a que se desarrolle algo más el mercado para tomar partido. La idea es permitir que se desarrolle al máximo este nuevo servicio distorsionando lo menos posible el mercado actual.

La respuesta a esta situación no es fácil. Si claramente se regula a favor de la VoIP, los operadores tradicionales, que también son proveedores de VoIP, podrían abandonar la antigua tecnología, lo que, no sólo les permitiría ahorros importantes de impuestos, sino que les permitiría desentenderse de sus obligaciones sociales de acceso universal, etc. Por otro lado, si se protege en exceso la tecnología tradicional se estaría frenando la evolución.

Otro hecho interesante en este panorama es que el desarrollo de la telefonía IP está íntimamente ligado al desarrollo de la banda ancha y, precisamente, son los operadores de telefonía quienes más están impulsando este despliegue. Sin embargo, estos operadores adquieren la mayoría de sus ingresos del tráfico de voz de la telefonía tradicional, al menos por ahora.

En general, la política que están siguiendo las administraciones es la de intervenir lo mínimo posible. No obstante, los propios hechos están forzando a tomar determinadas decisiones reguladoras.

Por ahora, parece evidente que un servicio de VoIP es claramente un servicio de datos mientras que sus participantes sean usuarios de Internet y no utilicen numeración de red telefónica. La cosa cambia cuando uno de sus participantes es usuario de una red de telefonía, o cuando se utiliza una numeración de red telefónica.

En cualquier caso, los retos con los que se encuentra el regulador son los siguientes:

- Garantizar el desarrollo de la nueva tecnología de voz sobre IP.
- Garantizar que este nuevo servicio de telefonía se interconecte con las redes telefónicas tradicionales.
- Garantizar que los usuarios pueden llamar a los números de emergencia (112 en Europa, 911 en Norteamérica, etc.).
- Gestionar adecuadamente aquellos recursos públicos que, por su naturaleza, son limitados: numeraciones, espectro radioeléctrico, etc.

Este último punto trae a colación un tema importante, la numeración. Quizás en un futuro la telefonía utilice un sistema de identificación del usuario más avanzado, amigable y flexible que el antiguo sistema de numeración telefónica a que estamos acostumbrados, pero, actualmente, la única forma de hacer compatible la telefonía IP con la telefonía tradicional es integrando la primera en el sistema de numeración de la segunda. El inconveniente es que este sistema de numeración responde a una distribución geográfica de tarifas, mientras que la telefonía IP permite el nomadismo (movilidad).

En el caso de España, el regulador ha optado por asignar dos rangos de numeración a la telefonía IP:

- Rango de numeración específica (51). Los números que empiecen por 51 se corresponderán con teléfono IP sin limitación geográfica (nómadas).
- Rango de numeración geográfica (8). Este bloque de numeración será compartido con el servicio telefónico tradicional y será asignado al servicio de telefonía IP con movilidad limitada (dentro del distrito de tarificación donde resida el abonado).

En ambos tipos de numeración se exige a los operadores el encaminamiento gratuito de las llamadas al número de emergencia 112. En el caso de la numeración geográfica, el centro de atención de las llamadas 112 será el correspondiente al de su distrito telefónico, mientras que en el caso de la numeración específica será el que designe el cliente en el momento de su contratación.

PAÍS	NUMERACIÓN
Alemania	032
Austria	0780
España	51
Francia	087
Holanda	85
Noruega	85
Reino Unido	056
Suecia	075

Tabla 1.1. Ejemplo de numeraciones específicas de VoIP en Europa

Puede obtenerse más información sobre regulación en el Grupo Europeo de Organismos Reguladores Independientes, IRG, en el Instituto Europeo de Estándares de Telecomunicaciones, ETSI (*www.etsi.org*), en la Unión Internacional de Telecomunicaciones, UIT (*www.itu.int*), o en la Asociación de Organismos Reguladores de América Latina, Regulatel (*www.regulatel.org*).

La evolución tecnológica

LA INVENCIÓN DEL TELÉFONO

Todos estudiamos que el inventor del teléfono fue Alexander Graham Bell. No obstante, este es un tema muy discutido últimamente. Según parece, la segunda mitad del siglo XIX fue muy activa en intentos de desarrollar un aparato que fuese capaz de transmitir la voz a distancia. La primera persona en darse cuenta que esto era posible fue el telegrafista francés Charles Bourseul. El 26 de agosto de 1854 publicó en L'illustration de París una nota en la que decía: *"Imaginad que se hable cerca de una placa metálica bastante flexible para que no se pierda ninguna de las vibraciones producidas por la voz; imaginad que esta placa establezca e interrumpa sucesivamente la comunicación con una pila; se podrá tener a distancia otra placa que ejecutará exactamente las mismas vibraciones"*.

Bourcel se quedó en la teoría. El primer intento práctico del que se tiene conocimiento se realizó en 1861. Ese año, el alemán Johann Philipp Reis construyó un aparato, al que llamó teléfono, que era capaz de transmitir sonido a distancia. Desgraciadamente, ese aparato tenía una escasa eficiencia, por lo que nunca alcanzó distancias superiores a los 100 metros. Además, aunque se hicieron pruebas en las que se transmitía sonido, nunca se demostró que fuese capaz de transmitir voz de forma inteligible. A pesar de ello, el 22 de marzo de 1876, una editorial de The New York Times, titulada *The telephone*, le otorgaba la invención del teléfono. Por cierto, en las pruebas realizadas, se dice que la primera frase que se transmitió fue: *"Das Pferd frisst keinen Gurkensalat"* (El caballo no come ensalada de pepino).

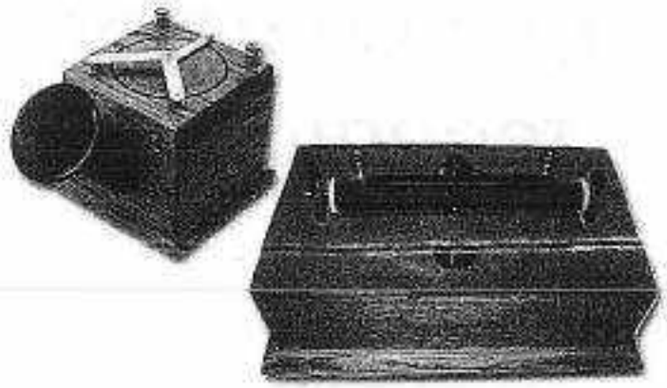

Figura 2.1. Emisor y receptor del teléfono de J. P. Reis (1862)

El siguiente capítulo de la historia es algo más complejo. Por una parte interviene Antonio Meucci, un emigrante italiano con gran ingenio para la técnica, pero con pocas cualidades empresariales o para los idiomas. Una anécdota, esta última, que marcó claramente su destino. Por otro, un emigrante escocés, Alexander Graham Bell, con buena posición social y habilidad para los negocios. Aunque ambos vivían en Estados Unidos en el momento de sus invenciones, sus distintas nacionalidades de origen influyeron para que cada uno de ellos tuvieran defensores y detractores independientemente de los hechos reales ocurridos.

El resultado fue que Alexander Graham Bell se llevó la patente, la fortuna y la gloria. No obstante, parece que la historia le está dando algo de esto último a Antonio Meucci al empezar a reconocerlo como el verdadero creador del primer aparato telefónico que permitía transmitir la voz a larga distancia.

En cualquier caso, la polémica no ha acabado. Para los franceses, el inventor del teléfono seguirá siendo el primero que expuso la teoría: Charles Bourseul. Para los alemanes será el primero que construyó un aparato llamado teléfono: Johann Philipp Reis. Para el mundo anglosajón no cabe duda que el inventor del teléfono fue Alexander Graham Bell, mientras que para los italianos y el Congreso de los Estados Unidos el merecedor de la gloria es Antonio Meucci.

La historia se rescribe. Antonio Meucci

Antonio Meucci nació en abril de 1808 en San Frediano, un pueblo cerca de Florencia (Italia). Allí estudió Ingeniería Mecánica. En 1835 emigró a Cuba, donde trabajaba como diseñador y técnico de escenarios en el teatro Tacón de La Habana. No obstante, su gran pasión era la ciencia y la experimentación, lo que le llevó a desarrollar una gran variedad de inventos. Uno de ellos, que llegó a alcanzar una gran popularidad en su tiempo, fue un método de descargas eléctricas para tratar enfermedades. Fue entonces cuando descubrió que se podía transmitir la voz utilizando la electricidad.

Después de darle vueltas a la idea durante diez años, se cree que consiguió construir su primer aparato telefónico en 1849. En 1850 emigró a Estados Unidos, donde se estableció en Clifton, cerca de Nueva York. En un entorno de inmigrantes italianos desarrolló nuevos inventos y los utilizó en distintas aventuras empresariales que nunca proliferaron.

En 1855 perfeccionó un sistema telefónico que utilizaba para comunicar el dormitorio de su mujer inválida (padecía un fuerte reumatismo) situado en una segunda planta con su laboratorio de la planta baja. Consciente de la importancia de este invento, al que llamó teletrófono, intentó captar capital para su comercialización. En 1860 organizó demostraciones públicas, e incluso, publicó una descripción de su aparato telefónico en un periódico italiano de Nueva York. Pero la prometida financiación nunca llegó.

En 1871 hizo una inscripción preliminar de patente (*caveat*, en su terminología en inglés) que renovó en 1872 y 1873, pero nunca pudo reunir la suma de 250 dólares para hacer el registro de patentes definitivo. De hecho, no pudo ni reunir 10 dólares para renovar la inscripción preliminar en 1874. Para más desgracias, un accidente le mantuvo largo tiempo en el hospital, lo que obligó a su esposa a vender varios de sus prototipos, incluido su primer teléfono, por 6 dólares.

En sus esfuerzos por rentabilizar el invento en el que había invertido tanto tiempo, le pidió al presidente de la recién establecida Western Union Telegraph Co que le dejara demostrarle las posibilidades que tenía su aparato parlante en la red de telégrafos de su compañía. Lo cierto es que nunca lo recibieron ni le devolvieron los prototipos que les envió. Le argumentaron que se habían perdido, pero, curiosamente, Alexander Graham Bell trabajaba en el laboratorio de Western Union donde, supuestamente, se habían guardado estos modelos.

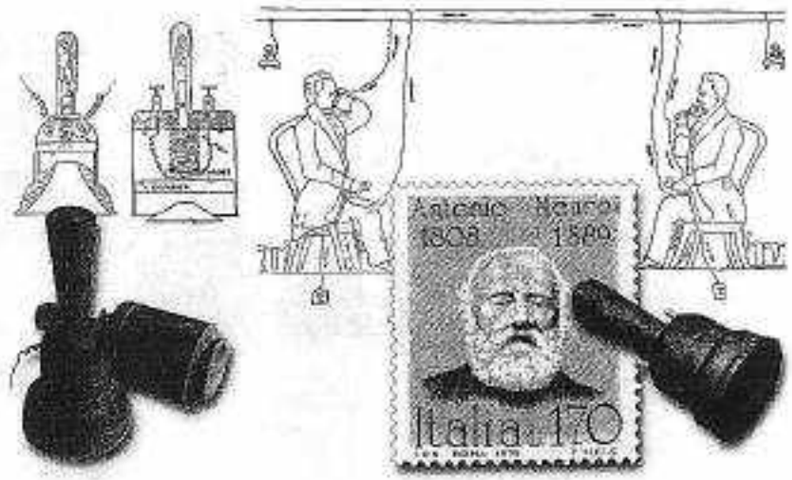

Figura 2.2. El teléfono de A. Meucci y dibujos de su potente (1871).

Bell hizo una inscripción preliminar de patente el 14 de febrero de 1876, que, aunque no describía el teléfono, se refería a él. Meucci contactó con un abogado para protestar por este hecho, pero, misteriosamente, toda la documentación de registro de patentes que había realizado Meucci en los años anteriores había desaparecido. El caso llegó a los tribunales y en 1886 el Secretario de Estado llegó a confirmar que había suficientes pruebas para otorgar el invento del teléfono a Meucci. El propio Thomas Alva Edison (quien patentara el invento de la bombilla pocos años después) envió una carta al juez posicionándose a favor de Meucci.

De hecho, en 1887 el Gobierno de Los Estados Unidos le anuló la patente a Bell. Sin embargo, la resolución definitiva del juicio se fue posponiendo año tras año hasta que Meucci murió en 1896 y el caso se cerró tal cual. El rico Bell le ganó la batalla al pobre Meucci. Ha tenido que pasar más de un siglo para que el Congreso de los Estados Unidos le reconociera a Meucci su invención. La resolución 269 del 11 de junio de 2002 dice: *"La vida y obra de Antonio Meucci deben obtener un justo reconocimiento, y debe serle atribuida la invención del teléfono"*.

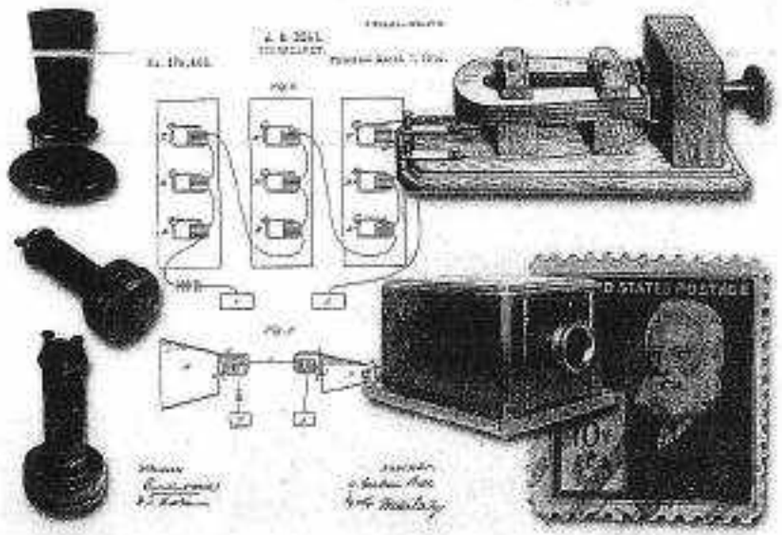

Figura 2.3. Dibujo en la patente y modelos de teléfonos de A.G. Bell (1877)

El despegue de la telefonía. Graham Bell

Alexander Graham Bell se esforzó en mejorar el invento que había patentado recientemente. En octubre de 1876 consiguió establecer una transmisión de voz a una distancia de 2 millas (más de 3 kilómetros). A él se unieron otros inventores de la época, quienes aportaron mejoras en los equipos y sistemas de transmisión. Thomas Alva Edison patentó un transmisor de voz en 1877.

En 1877, Graham Bell, junto con Sanders y Hubbard, crearon la primera empresa de telefonía, la Bell Telephone Company. Se dedicaron a alquilar teléfonos y franquiciar el servicio. Unos meses después, Western Union decidió introducirse en el negocio de la telefonía con unas patentes distintas a las de Bell. Curiosamente, el año anterior, el

presidente de Western Union había declarado en una nota interna de la compañía: *"El teléfono tiene muchos defectos como para poder considerarlo seriamente como un medio de comunicación"*.

Las primeras líneas telefónicas se contrataban por pares. El cliente seleccionaba qué dos puntos quería interconectar. La primera centralita comercial se puso en funcionamiento en enero de 1878 en New Haven, Connecticut, atendía a 21 teléfonos. Un par de meses después aparecería el primer directorio telefónico, tenía 50 nombres. No obstante, la primera patente de centralita la registró Leroy Firman en 1882.

En 1879, Graham Bell le ganó el juicio a Western Union por infringir su patente, lo que hizo que Western Union se tuviese que retirar de este negocio, dando paso a que se creara una gran compañía telefónica: la American Bell Company con 133.000 clientes. En 1885 se crearía una nueva compañía para prestar los servicios de larga distancia a la compañía de Bell. A esta nueva compañía se la llamó AT&T (*American Telephone and Telegraph*).

Hasta 1880, las líneas telefónicas se construían con un solo hilo. El retorno se hacía a través de tierra. Este sistema producía mucho ruido e interferencias que se corrigieron al empezar a utilizarse circuitos de dos hilos.

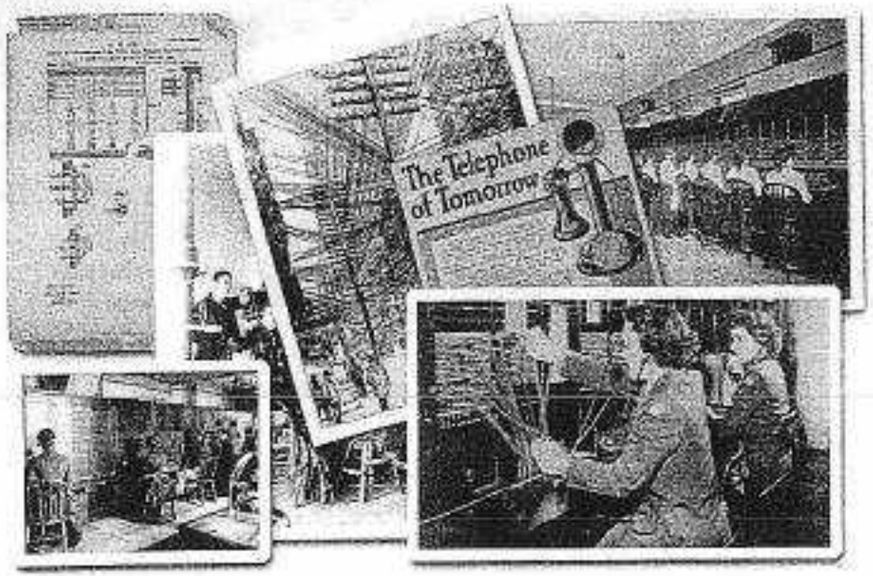

Figura 2.4. Historia del teléfono

En 1889 apareció el primer teléfono público de moneda en Hartford, Connecticut. La primera central telefónica automática data de 1889. Podía conectar hasta 100 líneas y fue patentada por Almon Strowger en 1891. Su primera instalación comercial se hizo en 1892 en La Porte, Indiana. Para ese año, la compañía de Bell contaba con más de 240.000 abonados.

Por cierto, con esta central de Strowger había que realizar la marcación pulsando un botón el número de veces deseado. Un socio de Strowger inventaría el disco de marcación en 1896.

Con el cambio de siglo, la compañía American Bell Company contaba con 800.000 clientes. Además existían multitud de pequeñas compañías independientes que contaban con unos 600.000 clientes. A pesar de ello, curiosamente, el desarrollo del teléfono fue mucho más lento que el de la telegrafía. Después de todo, para qué quiere alguien hablar con una persona distante.

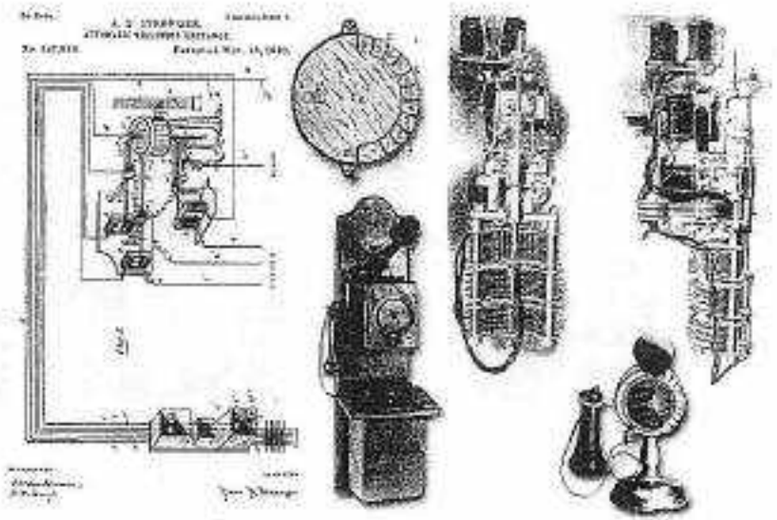

Figura 2.5. Algunas aportaciones de A. Strowger a la telefonía automática

EL CAMBIO TECNOLÓGICO

Las redes telefónicas se componen, fundamentalmente, del propio aparato telefónico conectado mediante redes de cableado urbano a las centrales telefónicas

locales (conocidas en inglés como *central office* o *local exchange*). Por otro lado, todas las centrales locales están interconectadas entre sí mediante centrales intermedias de tránsito (conocidas en inglés como *tandem office* o *toll office*).

Las centrales telefónicas constan de un equipo de conmutación, que es el que selecciona el circuito por el que se establece la llamada, y de equipos de transmisión, que es el que transmite la señal al resto de centrales con las que está interconectada.

Durante el siglo XX, la mecánica de los primeros equipos de conmutación fue sustituida por mecanismos eléctricos, electrónicos y finalmente por circuitos digitales. De la misma forma, los sistemas de transmisión analógicos, con cable coaxial o radio, fueron sustituidos por sistemas digitales sobre fibra óptica o radio digital.

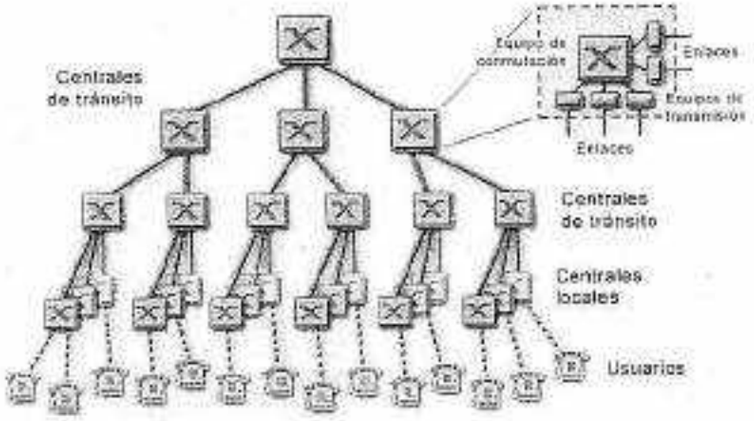

Figura 2.6. Red telefónica conmutada

A pesar de ello, la evolución de la red de telefonía ha tenido algo en común: se trata de una **red de conmutación de circuitos**. Esto quiere decir que para realizar una comunicación se van interconectando circuitos de los distintos tramos de redes hasta que entre uno y otro participante queda establecida una conexión directa dedicada en exclusividad. Se puede decir que entre ambos extremos existe un circuito ininterrumpido que los interconecta durante todo el tiempo que dura la comunicación, independientemente de que se hable o no. En este escenario es lógico que las comunicaciones telefónicas se cobren por tiempo de duración de la llamada.

A la red telefónica se le ha venido a llamar red telefónica conmutada (RTC), red telefónica básica (RTB), o por sus siglas en inglés: PSTN (*Public Switched Telephone Network*, 'Red pública telefónica conmutada') o GSTN (*General Switched Telephone Network*, 'Red telefónica conmutada general'). Adicionalmente, a la RTB que dispone de sistemas digitales de transmisión entre centrales se le dio el nombre de RDI (Red Digital Integrada).

A pesar de que las centrales telefónicas y los medios de transmisión evolucionaron hacia la tecnología digital, mejorando considerablemente la calidad de las comunicaciones e introduciendo nuevos servicios, la conexión de la central telefónica local con el usuario seguía siendo analógica. Para completar la digitalización apareció a finales de los años ochenta una nueva tecnología conocida como Red Digital de Servicios Integrados (ISDN, *Integrated Services Digital Network*) que permite la interconexión digital extremo a extremo, de terminal a terminal, tanto para voz como para datos. De hecho, uno de los usos más comunes de la RDSI ha sido el establecimiento de videoconferencias.

La tecnología digital

Como se ha visto anteriormente, mientras que las señales analógicas varían de forma continua, pudiendo tomar cualquier valor dentro de su rango, las señales digitales sólo toman determinados valores. Cuando la señal digital sólo toma dos valores (representados por 0 y 1) se trata de una señal digital binaria.

Los sonidos son movimientos ondulatorios del aire que son captados por el oído y transmitidos al cerebro, que los interpreta. El sonido, y por tanto la voz, por naturaleza, es analógico. Los primeros teléfonos utilizaban un micrófono formado por gránulos de carbón para convertir las variaciones de la presión de las ondas sonoras en variaciones del flujo de la corriente eléctrica. Una unidad similar en el otro extremo convierte las variaciones de corriente en una vibración de la membrana que produce ondas de presión acústica fieles a las originales.

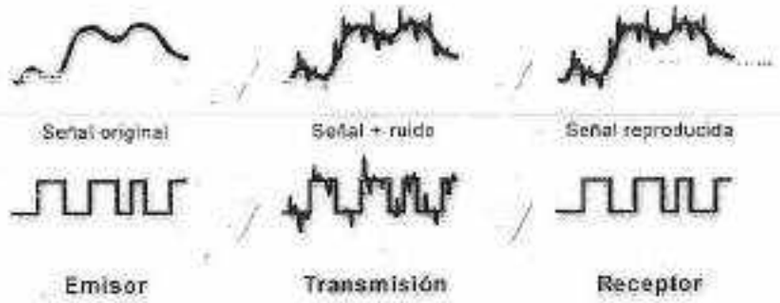

Figura 2.7. Ventajas de la tecnología digital

El gran inconveniente de las señales analógicas es que se ven afectadas, con relativa facilidad, por los ruidos introducidos por el entorno y por los propios sistemas empleados en su transmisión. Los ruidos e interferencias son de naturaleza analógica, igual que la señal, con lo que en el extremo distante no es fácil distinguir qué parte de la señal recibida se corresponde con la original y qué otra parte se corresponde con el ruido. Por el contrario, las señales digitales tienen la gran ventaja de ser muy resistentes al ruido, ya que en el extremo distante se pueden identificar muy bien los ceros y los unos originales y separarlos de la señal analógica del ruido.

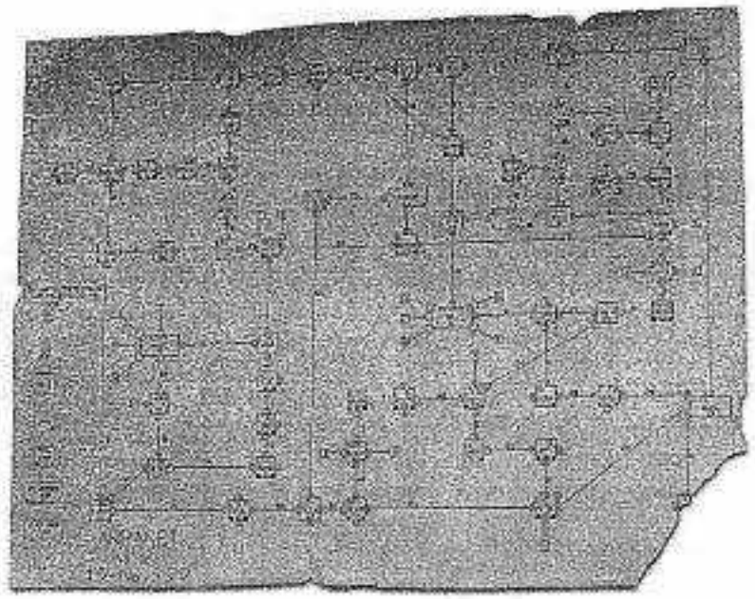

Figura 2.8. Plano de Arpanet de 1979

La conmutación de paquetes

La historia de la conmutación de paquetes comenzó a finales de los años sesenta. La Unión Soviética había lanzado el satélite Sputnik en 1957, se estaba en plena guerra fría, y Estados Unidos quería estar seguro de estar a la cabeza de la tecnología militar. En ese ambiente, el Departamento de Defensa de Estados Unidos cayó en la cuenta de que la tecnología empleada por la tecnología tradicional, la conmutación de circuitos, era demasiado frágil para resistir el más mínimo ataque, y mucho menos la tan temida guerra nuclear. Si se destruía una conexión entre dos centrales importantes o quedaba una central fuera de servicio, buena parte de las telecomunicaciones de defensa podrían quedar inutilizadas.

Este hecho llevó a que, en 1969, los señores Bolt, Beranek y Newman diseñaran y desarrollaran una red experimental conocida como Arpanet. La idea era conseguir una red con una tecnología tal que asegurase que la información llegara al destino aunque parte de la red quedase destruida. Esta nueva tecnología, propia de las comunicaciones digitales, se conoce con el nombre de **conmutación de paquetes**.

Mediante la conmutación de paquetes, toda la información que sale de un terminal origen es troceada en bloques de una determinada longitud, llamados paquetes. A cada paquete se le añade una información adicional al comienzo del mismo, llamada cabecera, con toda la información necesaria (por ejemplo, la identificación del terminal de destino o el número de secuencia del paquete) para que se pueda mover por la red de forma independiente y encontrar el camino para llegar al destino. Si en un momento dado queda bloqueada una ruta o un equipo de comunicaciones, los paquetes que iban a utilizar esos medios son enrutados de forma automática por caminos alternativos sin que se interrumpa la comunicación. Arpanet se convertiría en Internet unos años después.

Un beneficio adicional de las redes de conmutación de paquetes es que consiguen una utilización mucho más eficiente de sus recursos.

Comparación de tecnologías

Una línea telefónica es el par de hilos de cobre que llega a casa desde la central, y por extensión, también recibe este nombre el propio servicio telefónico. Desde el punto de vista técnico, una línea telefónica puede transmitir cualquier frecuencia comprendida entre los 300 y los 3.400 Hz., lo que quiere decir que el ancho de banda es de 3.100 Hz (3.400-300 Hz). Para hacernos una idea, el ancho de banda del télex es de 120 Hz y el de un canal analógico de televisión de 5 MHz.

La voz humana emite frecuencias que pueden llegar a los 10 KHz, pero como la línea telefónica corta todas las frecuencias superiores a los 3.400 Hz, este es el origen de que la voz por teléfono tenga ese sonido tan particular.

No obstante, la línea telefónica se utiliza también para ofrecer otros servicios, como son el fax o la transmisión de datos. En estos casos, un ancho de banda de 3.100 Hz limita la velocidad de transmisión de datos a 56 Kbps. Esto quiere decir que, aunque la línea telefónica ofrezca una calidad de sonido excelente, tiene grandes limitaciones para la transmisión de datos.

Las redes telefónicas se han desarrollado pensando en transmitir la señal de la voz. Esto quiere decir que la señal llega al destino prácticamente sin retardo y en el mismo orden como sale del origen.

Las redes de datos, como IP, tienen unos requerimientos distintos: por un lado admiten un mayor retardo, incluso, no importa si unos paquetes tardan más en llegar y otros menos; por otro lado, si el orden de la información se altera durante la transmisión, siempre se puede recomponer en destino sin problemas. Los pequeños retardos de las redes de datos (del orden de milisegundos) no tienen la más mínima importancia a la hora de recibir un correo electrónico o de ver una página Web. Lo que sí tiene gran importancia para las redes de datos es que llegue al destino absolutamente toda la información, sin que se pierda nada. Para la voz, sin embargo, lo importante es que llegue la información en tiempo real, no importando que se pierdan pequeños fragmentos mientras que el resultado siga siendo comprensible (se admiten pérdidas de hasta el 5%).

EL RETO DEL TIEMPO REAL EN IP

Las redes de datos, como Internet, no fueron pensadas para permitir el intercambio de información en tiempo real. Su finalidad siempre ha sido el intercambio de datos con total seguridad, fiabilidad y ahorro de recursos. Por tanto, nos encontramos ahora con el reto de poder realizar una comunicación en tiempo real con una red que no fue pensada para eso.

Las redes de telefonía tradicional son un tipo de red conocida como *orientada a la conexión*. Esto quiere decir que, antes de que comience la comunicación se establece un circuito, una conexión, entre el origen y el destino. Cuando un usuario levanta el teléfono, la red le envía un tono invitándolo a marcar el número del destinatario. Cuando el destinatario descuelga la red establece una conexión extremo a extremo que permanecerá abierta durante todo el tiempo que dura la comunicación, independientemente de que haya o no intercambio de información. Cuando uno de los participantes cuelga, la red libera los recursos utilizados para poder ser utilizados por cualquier otro usuario.

Una de las desventajas de este proceso es que se ocupan recursos que pueden no estar siendo utilizados. Por el contrario, una de sus ventajas es que, una vez establecida la comunicación, sus características técnicas (retardo, etc.) permanecen invariables. Las redes orientadas a la conexión, como la red telefónica tradicional, son redes confiables.

Las redes de datos se clasifican como redes *no orientadas a la conexión*. A cada paquete se le añade la información suficiente como para que llegue al destino por sus propios medios. Es algo así como cuando se echa una carta al correo. En el sobre se incluye la información necesaria para que la carta llegue a su destino, pero, lo más que podemos hacer en ese momento es confiar que eso ocurrirá en un tiempo razonable.

Realmente no sabemos el camino que seguirá la carta, ni tampoco sabemos cuánto durará el viaje. De hecho, la carta podría perderse, en esta caso, el destinatario nos solicitaría que se la volviésemos a enviar. El hecho es que no hay una conexión directa en tiempo real entre los extremos de la comunicación. La transferencia de archivos, el correo electrónico o el servicio Web funciona muy bien con este sistema, pero la voz necesita tener una velocidad de transmisión constante y con el menor retardo posible.

La pregunta es: cómo hacemos que una aplicación orientada a la conexión (como la voz o el vídeo) funcione en un entorno que no está orientado a la conexión. Para lograrlo se ha mejorado el protocolo IP de forma que corrija estos defectos. Estas mejoras incluyen lo siguiente:

- **Protocolo de transporte en tiempo real** (RTP, *Real-time Transport Protocol*, RFC 3350). Protocolos que hacen posible que la información llegue al destino con la misma cadencia de tiempo con la que se introduce en el origen y con un retardo mínimo.
- **Protocolo de control de RTP** (RTCP, *RTP Control Protocol*, RFC 3350). Realiza un seguimiento de la calidad de la conexión RTP.
- **Protocolo de reserva de recursos** (RSVP, *Resource Reservation Protocol*, RFC 2205). Realiza una asignación de recursos para asegurar el ancho de banda necesario entre origen y destino.
- **Protocolo de retransmisión en tiempo real** (RTSP, *Real-Time Streaming Protocol*, RFC 2326). Asegura el envío de flujos de datos en tiempo real.
- **Protocolo *multicast*** (RFC 1112 y 2236). Este protocolo permite que la información generada en un punto se puede transmitir a múltiples destinos sin tener que reenviarla a cada uno de ellos. Este sistema es útil para el servicio de conferencias.
- **Protocolo de descripción de la sesión** (SDP, *Session Description Protocol*, RFC 2327). Facilita el intercambio de las características de la emisión.
- **Protocolo de anuncio de sesión** (SAP, *Session Announcement Protocol*, RFC 2974). Son paquetes que se envían periódicamente para identificar las sesiones que están abiertas y que pueden ser de interés para el usuario final.

El problema de la estandarización

Bueno, ya sabemos qué es lo que hay que hacer: crear un conjunto de protocolos que funcionen sobre IP y que permitan el intercambio de datos en tiempo real. El problema está en lograrlo de forma que todos los dispositivos que se fabriquen sean interoperables. Esto es, el problema es la estandarización.

Típicamente, cada fabricante ha desarrollado sus propias soluciones que funcionaban exclusivamente con sus propios productos. Se puede decir que el comienzo de la VoIP ha sido más competitivo que colaborador. Para garantizar la interoperatividad, los organismos de normalización han desarrollado diferentes estándares. En el mundo de la VoIP existen dos organismos claves: La Unión Internacional de Telecomunicaciones (UIT o ITU, *International Telecommunications Union*), con sede en Ginebra y la Sociedad Internet (ISOC, *Internet Society*).

La UIT viene siendo responsable de la normalización de los sistemas de comunicaciones telegráficos, telefónicos, de radio y televisión. Sus trabajos se dividen en tres sectores: el sector de las radiocomunicaciones (UIT-R), el de las telecomunicaciones (UIT-T), y el de desarrollo de las telecomunicaciones (UIT-D). El responsable de la normalización de la VoIP es UIT-T, el cual ya desarrollaría otros estándares como el de la RDSI (ISDN) o el ATM (*Asynchronous Transfer Mode*, 'Modo de transferencia asíncrona' o MTA). Los documentos que describen los estándares se nombran con una letra, que identifica un área tecnológica concreta, seguido de un número que identifica cada estándar particular. Los estándares que comienzan con la letra H se refieren a los sistemas multimedia y audiovisual. El estándar más conocido relacionado con la VoIP es el H323, cuyo título oficial es Sistemas de comunicaciones multimedia basado en paquetes.

Año	Evento
1989	*Alon Cohen y Lior Haramaty fundan Vocaltec*
1995	*Sale al mercado Internet Phone y Cu-Seeme*
1996	*Telecom Finland ofrece VoIP de PC a PC*
1996	*Nace Microsoft Netmeeting que ofrece VoIP de PC a PC*
1996	*Net2Phone ofrece comunicaciones telefónicas de PC a teléfono*
1996	*Se publica el estándar RTP/RTCP*
1996	*Se publica el estándar H.323*
1997	*Se publica el estándar RSVP (RFC2205)*
1997	*Deutsche Telecom ofrece VoIP de teléfono a teléfono*
1998	*Se construyen los primeros gateways y gatekeepers*
1999	*Se publica el estándar SIP*
1999	*Se publica el estándar MGCP*
2001	*Messenger sustituye a Netmeeting*
2001	*Vonage ofrece servicios de VoIP para empresas*
2001	*Se publica el estándar MPLS*
2003	*Nace Skype de la mano de Niklas Zennström y Janus Friis*
2005	*eBay compra Skype por 2.100 millones de euros*

Tabla 2.1. Hitos en el desarrollo de la telefonía IP

Por su lado, la Sociedad Internet, ISOC, se creó en 1992 para poner orden en las tecnologías relacionadas con Internet. ISOC está organizada en distintos grupos de trabajo entre los que se encuentra el Grupo de Ingeniería de Internet (IETF, *Internet Engineering Task Force*). El IETF es responsable de publicar los estándares de Internet, conocidos como documentos RFC (*Request For Comments*, 'Solicitud de comentario'). Cada documento RFC se identifica por un número. El relacionado con la VoIP es el RFC 3261, también conocido como Protocolo de inicio de sesión o SIP (*Session Initiation Protocol*).

Es curioso el hecho de que, mientras que el origen de la UIT es la conmutación de circuitos, el origen de ISOC es la conmutación de paquetes. Esta diferencia se ha dejado ver en las distintas aproximaciones que han tenido a la hora de definir sus correspondientes estándares.

Por tanto, tenemos dos estándares principales que definen cómo debe Internet manejar los paquetes que transportan voz: H323 y SIP.

LOS COMIENZOS DE LA VOZ SOBRE IP

Los primeros años de Internet pasaron en un entorno militar y universitario casi exclusivamente. Con los años 90, la red fue evolucionando hacia un entorno cada vez más comercial, a la vez que veía cómo sus usuarios crecían de forma exponencial. No obstante, en esa época los usuarios solían conectarse a Internet utilizando una línea telefónica y un módem. En ese entorno no parecía tener sentido digitalizar la voz para transmitirla por Internet, ya que para acceder a Internet se utilizaba, precisamente, la red telefónica. Por otro lado, la mayoría de los PC no disponían de tarjeta de audio, eran ordenadores sordos y mudos.

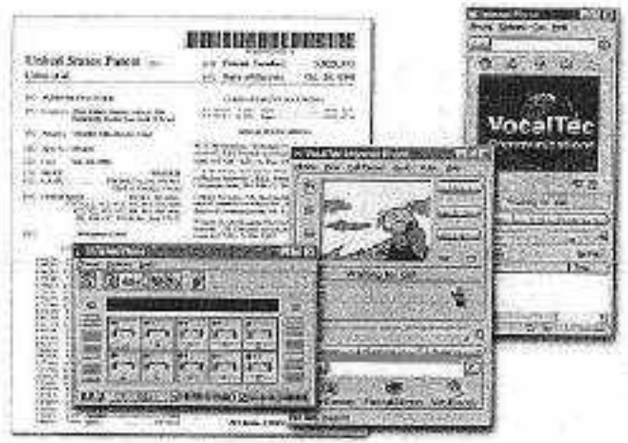

Figura 2.9. Patente del Audio Transceiver y versiones de Internet Phone

A pesar de ello, un ingeniero israelita llamado Alon Cohen desarrolló un dispositivo, al que llamó Audio Transceiver, que permitía que la voz ya digitalizada se adaptase al ambiente hostil de la red Internet (pérdida de paquetes, fluctuación de retardo, etc.). En 1989 fundó la empresa Vocaltec junto con Lior Haramaty y en 1995 sacó al mercado el primer producto comercial de VoIP: Internet Phone. Pocos meses después, en agosto, le saldría un competidor: CU-SeeMe.

Internet Phone funcionaba exclusivamente entre usuarios de PC que tuvieran instalado su software y tuviera una tarjeta de audio, altavoz y micrófono. La calidad del sonido era tan mala que muy pocos vieron esto como una alternativa real a la telefonía tradicional, pero demostró que era posible transmitir la voz por Internet. Vovaltec saldría exitosamente a bolsa en febrero de 1996, de donde obtuvo 33 millones de dólares. Ese mismo año se fundaría Net2Phone, la primera empresa que ofreció comunicaciones telefónicas desde PC a teléfonos tradicionales.

El primer operador de telefonía tradicional que comercializó la VoIP fue Telecom Finland en 1996. Lo hizo utilizando el software de Volcaltec para establecer comunicaciones de PC a PC. En 1997, Deutsche Telekom sacó un servicio, al que llamó T-NetCall, que permitía realizar llamadas telefónicas de teléfono a teléfono, utilizando Internet.

Figura 2.10. El mundo de Skype

En 1998, fabricantes de equipos, como Cisco o Lucent, decidieron desarrollar dispositivos (*routers*) especialmente pensados para manejar la voz. También se construyeron los primeros *gateways* (pasarelas) que permitían hacer llamadas de PC a teléfono y viceversa. Ese año aparecieron algunas compañías en Estados Unidos que permitían a sus usuarios realizar, con sus propios aparatos telefónicos, llamadas telefónicas gratuitas de larga distancia a cambio de escuchar unos breves anuncios publicitarios al comienzo y finalización de la llamada.

Microsoft sacó en agosto de 1996 su conocido software Netmeeting, el cual permitía establecer comunicaciones de voz del tipo de PC a PC. Muchos usuarios obtuvieron sus primeras experiencias de VoIP con este programa. En el año 2001 este software evolucionaría hacia Messenger.

En enero de 2001, Vonage empezó a ofrecer servicios de VoIP para empresas. Desde entonces los servicios de VoIP han ido creciendo en el entorno empresarial. Muchas empresas utilizan VoIP en sus centros de atención al cliente.

En estos últimos años, el sueco Niklas Zennström y el danés Janus Friis han escrito una nueva página de la historia de la voz sobre IP. En agosto de 2003 sacaron al mercado la primera versión del programa Skype. Este programa permite establecer llamadas telefónicas gratuitas entre ordenadores y a coste local entre ordenador y teléfono fijo tradicional. Recientemente han sacado también una versión para videoconferencias. El hecho es que en octubre de 2005 la empresa eBay compró Skype por 2.100 millones de euros. Skype es el software de VoIP más extendido del mundo en la actualidad.

Digitalización de la voz

INTRODUCCIÓN

El sonido no es más que variaciones de la presión del aire. Estas variaciones de presión son captadas por los micrófonos y convertidas en señales eléctricas cuya amplitud es proporcional a la presión del aire. Tanto la señal sonora como la señal eléctrica resultante varían de forma continua, o lo que es lo mismo, son señales de tipo analógico.

Internet maneja internamente señales digitales binarias. Quiere esto decir que las señales eléctricas que manejan los equipos que forman parte de la red Internet sólo entienden de dos niveles de tensión o de amplitud de señal. Estas señales no varían de forma continua, sino que toman uno u otro valor. Uno de estos valores representa a la información 0 y el otro a la información 1.

Para que Internet pueda manejar cualquier información analógica (como la voz), es necesario que ésta se convierta previamente en digital. Este proceso se conoce como digitalización. Evidentemente, una conversión de analógico en digital suele venir acompañada de su correspondiente conversión inversa, de digital en analógico, cuando dicha información llega al destino.

Aunque las primeras conversiones de analógico en digital suponían simplemente tomar muestras periódicamente e ir convirtiendo el valor de la amplitud de cada muestra en información binaria, en la actualidad, dicho proceso se ha complementado con técnicas de codificación que permiten mantener unos buenos niveles de calidad de sonido utilizando un ancho de banda (velocidad de transmisión) mucho menores. La importancia de la codificación es tal, que a todo el proceso se le conoce también como **codec**, o codificación y decodificación de la información.

El proceso de codificación y decodificación viene emparejado. Quiere esto decir que cada técnica de codificación tiene su correspondiente técnica de decodificación. Para que el sistema funcione, ambos extremos de la comunicación deben contar con el mismo tipo de codec. Ahí es donde intervienen los organismos reguladores. La UIT-T es el organismo regulador que ha venido marcando las pautas de los distintos estándares de codec: desde el tradicional modelo G.711, aparecido en 1972 y utilizado por las redes de telefonía en su proceso digitalizador, hasta los más recientes modelos, como el G.729, que consigue elevados niveles de calidad con una baja velocidad de transmisión.

El resultado es que en el mercado existen distintos sistemas de digitalización, lo que hace que no todas las aplicaciones o dispositivos de VoIP sean compatibles entre sí. De hecho, en general, cada proveedor de servicio de VoIP ofrece conectividad exclusivamente con sus propios usuarios dentro de Internet y con los de la red telefónica.

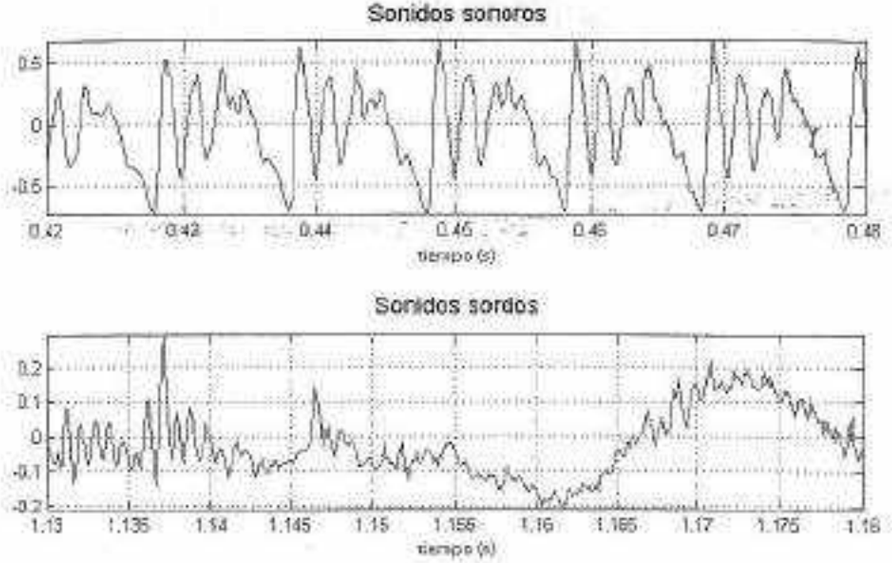

Figura 3.1. Segmentos típicos de voz

LA NATURALEZA DEL HABLA

El sonido de la voz no es mas que la vibración del aire producida por las cuerdas vocales. Aunque el oído puede percibir sonidos con frecuencias entre 20 y 20.000 Hz, la voz humana sólo es capaz de generar frecuencias entre 40 y 10.000 Hz.

El micrófono convierte las vibraciones sonoras en señal eléctrica analógica. Esta señal analógica es idéntica en frecuencia e intensidad al sonido que representa. Los primeros circuitos telefónicos transmitían la señal eléctrica tal cual, con lo que se utilizaba un par de hilos para cada circuito. Conforme la red fue creciendo surgió la necesidad de ahorrar costes, y se vio que no es necesario retransmitir todo el margen de frecuencias para que el sonido sea inteligible en destino. De hecho, se comprobó que se consigue el 100% de inteligibilidad retransmitiendo sólo las frecuencias menores de 5.000 Hz. A pesar de ello, en telefonía se ha admitido una cierta pérdida de inteligibilidad, reduciendo el margen de frecuencias que son retransmitidas a las comprendidas entre 300 y 3.400 Hz. Esta limitación le ha dado al sonido telefónico ese timbre tan característico. La llegada de la telefonía digital amplió el margen de frecuencias a los 4.000 Hz.

TIPO	FRECUENCIA FUNDAMENTAL (Hz)
Soprano (mujer)	260 a 1.040
Mezzosoprano (mujer)	220 a 880
Contralto (mujer)	200 a 780
Tenor (hombre)	130 a 520
Barítono (hombre)	100 a 390
Bajo (hombre)	80 a 330

Tabla 3.1. Ejemplos de frecuencias fundamentales de la voz

MEDICIÓN DE LA CALIDAD DE LA VOZ

Una de las características más importantes de los distintos sistemas de codificación y decodificación de la voz es la calidad de sonido inteligible conseguido en destino. La finalidad de la tecnología de voz sobre IP es conseguir una comunicación mediante el empleo del habla, por tanto, más que la fidelidad del sonido importa la inteligibilidad del mismo. Esto es lo que se conoce como calidad de la voz. La pregunta es ¿Cómo se puede medir un concepto que parece tan subjetivo?

En general, cuando se mide la calidad de la voz en comunicaciones telefónicas, no se habla de fidelidad del sonido, sino de estos tres parámetros básicos:

- Calidad de audición. Cómo se escucha, si se entiende bien.
- Calidad de conversación. Cómo se interactúa.
- Calidad de transmisión. Calidad de la red.

En cualquier caso, el objetivo de las mediciones es poder medir las mejoras de un sistema concreto o realizar comparaciones entre sistemas distintos. Para ello, suele ser suficiente con medir uno o dos de los parámetros anteriores.

Aunque parece una tarea complicada, el hecho es que existen diversos métodos normalizados para realizar mediciones de la calidad de la voz. Los más conocidos son: MOS, PSQM y PAMS

Método subjetivo. MOS

MOS (*Mean Opinion Score*, 'Valoración media de opinión') fue uno de los primeros métodos de medición de la calidad de la voz. Se trata de un sistema de valoración subjetiva que consiste en evaluar la calidad de distintas muestras y ponderar

los resultados para obtener una puntuación media. El sistema MOS dispone de una escala de cinco puntos con el siguiente significado:

1. **Mala**. Distorsión muy molesta
2. **Pobre**. Distorsión molesta
3. **Regular**. Distorsión perceptible y ligeramente molesta
4. **Buena**. Nivel de distorsión perceptible pero no molesta
5. **Excelente**. Sin distorsión

Para realizar una valoración se elige un grupo de personas (un mínimo de 30) que participan en una conversación empleando el sistema que se pretende evaluar o escuchan una selección de muestras de voces sometidas a dicho sistema. Cada participante valora la conversación o cada una de las muestras de acuerdo a la escala anterior. El resultado final asignado es el valor medio de las distintas valoraciones. Para la realización de las pruebas suelen utilizarse una selección de frases hechas (por ejemplo, las conocidas como frases de Harvard, *Harvard sentences*) que están pensadas para contener todos los sonidos que, típicamente, nos podemos encontrar en una conversación.

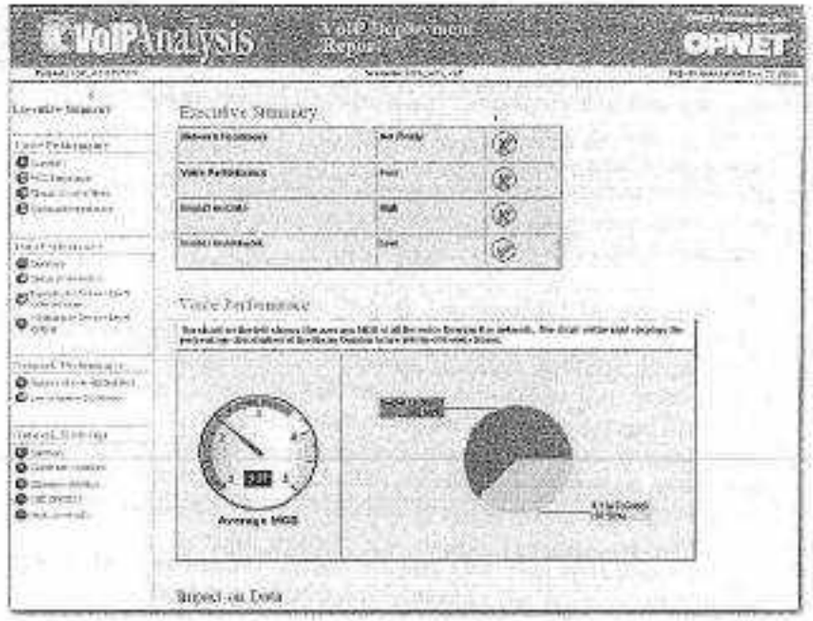

Figura 3.2. Software de análisis de VoIP

El sistema MOS está recogido en la recomendación P.800 de la UIT-T. Para asegurar en lo posible la objetividad de la medición, esta recomendación incluye una serie de indicaciones relativas a la selección de participantes, el entorno donde se deben realizar las pruebas, explicación de la valoración, análisis de resultado, etc. Idealmente, si se realizasen dos evaluaciones distintas a un mismo sistema debieran obtenerse puntuaciones similares.

Desde el punto de vista práctico, una puntuación MOS de 4 o superior supone una alta calidad de voz.

El gran inconveniente del sistema MOS es que su realización tiene un alto coste debido a los recursos y personas que deben involucrarse.

Métodos objetivos

En la idea de reducir los costes de evaluación de la calidad de la voz, la UIT-T sacó en 1998 una nueva recomendación, la P.861. Esta recomendación recoge las características del sistema conocido como PSQM (*Perceptual Speech Quality Measurement*, 'Medición perceptiva de la calidad del habla'), desarrollado por el Instituto KPN de Holanda.

La gran ventaja de este nuevo sistema es que se trata de un método objetivo que se lleva a cabo mediante el empleo de un dispositivo electrónico.

El sistema PSQM se basa en una comparación entre la señal a evaluar y la señal de origen. La prueba incluye una serie de algoritmos que determinan los efectos de un conjunto de variables como: la persona que habla (hombre, mujer o niño), el volumen de la voz original, el retardo, el porcentaje de silencio o el ruido ambiental. La puntuación final se obtiene después de realizar distintas mediciones. La escala de valores varía entre el 0, equivalente al valor MOS 5, y el 6,5, equivalente al valor MOS 1.

Recientemente, la UIT-T ha reemplazado la recomendación P.861 por la P.862, más conocida como PESQ (*Perceptual Evaluation of Speech Quality*, 'Evaluación perceptiva de la calidad del habla'). La escala PESQ se mueve en el rango de -1 a 4,5.

Otro de los sistemas objetivos de evaluación de la calidad de la voz es el conocido como PAMS (*Perceptual Analysis/Measurement System*, 'Sistema perceptivo de medida y análisis'). Este sistema pretende medir el nivel de esfuerzo necesario para comprender el habla. Para realizarlo se llevan a cabo distintas comparaciones entre el sonido original y resultante. El resultado es un valor en la escala de 1 a 5.

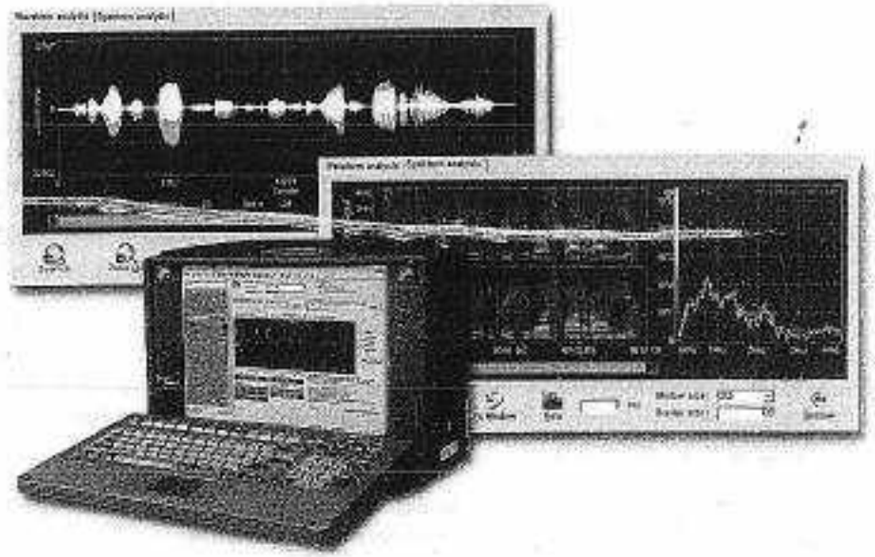

Figura 3.3. Medición de la calidad de la voz

A pesar de las grandes ventajas de las mediciones objetivas, el método MOS ofrece una mayor fiabilidad. Frecuentemente son utilizados los métodos objetivos en las prácticas de laboratorio y MOS como medición definitiva de calidad.

RETOS DE LA TRANSMISIÓN DE LA VOZ

Tradicionalmente, uno de los temas más críticos de la telefonía IP ha sido la calidad de la voz. Los primeros sistemas de voz sobre IP ofrecían una calidad de sonido claramente mejorable y, lo que era peor, esta calidad variaba en el tiempo. La razón es simple, la tecnología IP no fue diseñada para soportar el intercambio de voz o de cualquier otro tipo de señal interactiva en tiempo real (por ejemplo, vídeo). IP fue diseñado para la transmisión de datos, donde resulta extremadamente importante que no se pierda ni un solo bit, aunque es extremadamente tolerante con el retardo. La voz, sin embargo, es muy sensible al retardo, aunque no importa que algún bit no llegue al destino (se admite una pérdida de hasta el 5%). Por tanto, es claro que la transmisión de la voz tiene unos requisitos diferentes a la de los datos.

Retardo de ida y vuelta

En las comunicaciones de voz por satélite, la señal necesita 120 milisegundos para viajar de la tierra al satélite y otros tantos para realizar el camino de vuelta. A este tiempo hay que sumarle el tiempo necesario para que la señal se transmita por la red terrestre. El resultado es que una comunicación de voz que utilice un tramo satelital necesita entre 250 y 300 milisegundos en llegar de un extremo a otro.

Se llama retardo de ida y vuelta al tiempo necesario para que la señal de la voz haga el camino de ida y de vuelta entre los dos extremos de una comunicación. Dicho de otra forma, es el tiempo necesario para que un interlocutor escuche la respuesta a su mensaje. Al retardo de ida y vuelta también se le conoce como RTD, o *round trip delay*. En el caso del ejemplo del satélite, este retardo puede ser superior a 500 o 600 milisegundos. Cualquiera que haya mantenido una conversación por este medio conoce lo molesto que esto resulta.

La UIT establece que el retardo de ida y vuelta no debe superar los 300 milisegundos en las llamadas telefónicas (recomendación G.114). Como puede suponerse, en el caso de los datos no suele importar si un mensaje de correo tarda 3 segundos o 3 minutos en llegar al destino.

Fluctuación de retardo

Tan importante como el retardo en sí, es que dicho retardo permanezca constante. Cuando se tiene un retardo constante, las partes que intervienen en la comunicación se acaban adaptando a él (esto es lo que ocurre en las comunicaciones de voz por satélite). No obstante, si el tiempo de retardo varía constantemente, la conversación se vuelve mucho más incómoda, ya que genera en las partes una sensación de completo descontrol. Por cierto, a la fluctuación de retardo se le conoce habitualmente por el término anglosajón *jitter*.

La forma de evitar las fluctuaciones de retardo es utilizando memorias intermedias (búferes) que vayan añadiendo retardo cuando la comunicación es más rápida y quitándolo cuando es más lenta. El resultado es un retardo elevado, pero constante.

Como se puede suponer, las fluctuaciones de retardo no tienen ninguna consecuencia en la comunicación de datos. Cuando se recibe una página Web o un mensaje de correo, no influye que la velocidad de formación de la página o de recepción del correo no sea constante. Por tanto, la fluctuación de retardo es un inconveniente que afecta exclusivamente a las comunicaciones en tiempo real (voz y vídeo).

Los motivos de las fluctuaciones de retardo son dos:

- Enrutamiento variable. Los paquetes IP de una misma comunicación pueden tomar caminos distintos para llegar al destino, por tanto, pueden experimentar retardos distintos.
- Ocupación variable. Aunque todos los paquetes viajaran por la misma ruta, el nivel de ocupación de los equipos intermedios (*routers*) no es constante, por lo que el tiempo necesario para resolver los enrutamientos puede variar de unos paquetes a otros.

En el caso de la conmutación de circuito utilizada por las redes telefónicas tradicionales no existen fluctuaciones de retardo, ya que, una vez que se ha establecido el circuito que comunica ambos extremos, toda la información viaja por el mismo camino y utiliza los mismos recursos, que, además, están dedicados en exclusiva, se utilicen o no.

Pérdida de paquetes

Otro de los inconvenientes de las redes IP que pueden afectar a la calidad de la voz es la pérdida de paquetes. Esto es, de vez en cuando, algún paquete que forma parte de una comunicación se pierde en el camino. En las comunicaciones de datos, el equipo destinatario detecta la falta de este paquete y le pide al origen que se lo retransmita. El tiempo adicional necesario para esta retransmisión no suele afectar al resultado general de la comunicación.

En el caso de la voz, si se pierde un paquete resulta más conveniente darlo por perdido que esperar a la retransmisión del mismo. La espera genera un retardo indeseado, mientras que la falta esporádica de un paquete prácticamente no influye en el entendimiento de la comunicación, sea esta de voz o de vídeo.

TÉCNICAS DE CODIFICACIÓN

Existen tres tipos generales de técnicas de codificación y decodificación de la voz o codecs:

- De forma de onda
- Vocoder o de la señal origen
- Híbridos

Los codecs de forma de onda básicamente muestrean la señal analógica y la codifican directamente. Transmiten los valores cuantificados al extremo distante, donde se reconstruye la señal original. Este sistema de codec es más bien simple y reproducen una buena calidad de sonido en destino. Su mayor inconveniente es que, comparado con otros sistemas, necesitan un gran ancho de banda para conseguir unos mismos niveles de calidad.

El vocoder representa a la señal sonora de acuerdo a un modelo matemático. Periódicamente identifica el modelo con el que se corresponde el sonido de la muestra y envía al destino los parámetros que lo identifican. En el destino se va reconstruyendo la forma de onda de acuerdo a los parámetros recibidos. Es de destacar que, en el caso del vocoder, no se transmite la información directa de la voz, sino los parámetros que la definen. Este sistema puede verse completado con un identificador de silencio para conseguir una mayor eficiencia. Con el sistema vocoder se puede reducir la necesidad de ancho de banda hasta los 2,4 Kbps. En este caso, el sonido no es de buena calidad (suena como metálico) pero es inteligible.

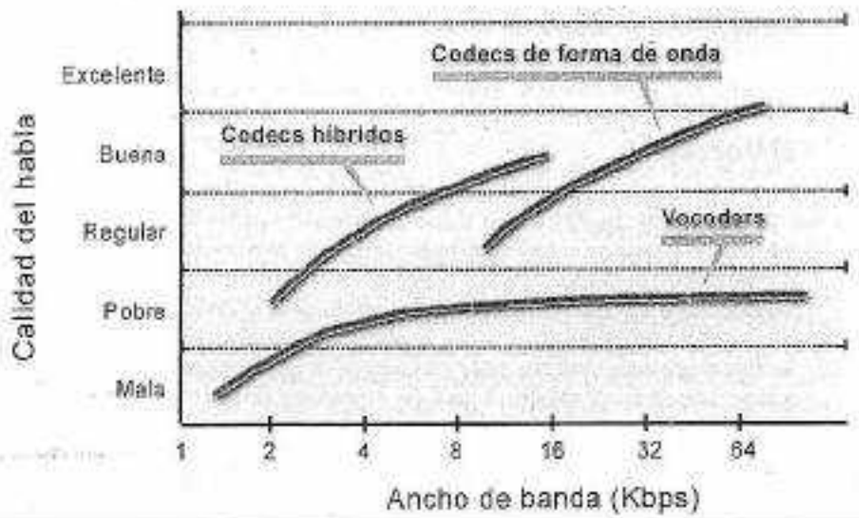

Figura 3.4. Comparación de los distintos tipos de codecs

El inconveniente del sistema vocoder es que no ofrece una alta calidad de sonido, ni aún aumentando considerablemente el ancho de banda. Para conseguir mejorar la calidad, se utilizan sistemas híbridos. Éstos ofrecen lo mejor de ambos sistemas anteriores, consiguiendo una muy buena calidad de sonido con un ancho de banda limitado.

Codec de forma de onda

Los codec de forma de onda realizan la digitalización de la señal analógica de la voz en dos pasos:

- Muestreo
- Cuantificación

El muestreo consiste en tomar muestras periódicas del nivel de la señal analógica para representar posteriormente estos valores en forma binaria mediante las técnicas de cuantificación. La información binaria es transmitida al destino donde se le aplica el proceso inverso para conseguir una señal analógica similar a la original.

FORMATO	MUESTREO	BITS/MUESTRA	MODO	ANCHO DE BANDA	FRECUENCIA
Teléfono	8 Khz	8	Mono	64 Kbps	200-3.400 Hz
Radio AM	11,025 Khz	8	Mono	88 Kbps	
Radio FM	22,050 Khz	16	Estéreo	705,6 Kbps	
CD	44,1 Khz	16	Estéreo	1411,2 Kbps	20-20.000 Hz
DAT	48 Khz	16	Estéreo	1536 Kbps	20-20.000 Hz

Tabla 3.2. Comparación de formatos de audio

Muestreo

Para que la señal conseguida en destino reproduzca fielmente la señal original es necesario que el tiempo entre muestra y muestra sea tal que no pueda ocurrir nada impredecible entre ellas. En 1928, el sueco Harry Nyquist desarrolló la hipótesis de que la señal puede ser reconstruida siempre que las muestras se tomen a una frecuencia doble de la frecuencia máxima de la señal original. Esta hipótesis fue demostrada matemáticamente, y convertida en teorema, por el estadounidense Claude Shannon veinte años más tarde (1948). A pesar de ello, el teorema se conoce como teorema de Nyquist. No obstante, Shannon, que murió hace poco (año 2001), es considerado el padre de la teoría de la información (de donde viene la Sociedad de la Información).

Volviendo al tema, si la máxima frecuencia de la señal analógica que se pretende retransmitir es 4.000 Hz, aplicando el teorema de Nyquist se deben tomar 8.000 muestras cada segundo, o lo que es lo mismo, una muestra cada 0,125 milisegundos.

Figura 3.5. Muestreo y cuantificación

Cuantificación

Una vez que sabemos cada cuánto tiempo tenemos que tomar muestras, la siguiente cuestión es cuántos bits son necesarios para representar este valor. Parece una pregunta simple, pero lo cierto es que cualquier valor digital sólo representa un conjunto de valores discretos. Por ejemplo, si el nivel de tensión máximo de la señal de la voz es 30 y utilizamos 2 bits para representarlo, sólo podremos transmitir 4 niveles de tensión (00, 01, 10 y 11). Estos niveles podrían corresponderse con los valores 0, 10, 20 y 30. Si en vez de 2 bits se utilizasen 3 o 10, el número de niveles de tensión representados aumentaría, pero seguirían siendo limitados y, por lo tanto, seguiría existiendo la necesidad de realizar una aproximación.

A la asignación del valor binario correspondiente a cada valor real de la muestra se lo conoce como cuantificación. Como la cuantificación siempre se lleva a cabo por aproximación, esto quiere decir que existe un error (diferencia entre la señal original y la cuantificada) y, por tanto, la señal de destino será similar a la original, pero no idéntica. A este error se lo conoce como error o ruido de cuantificación.

Cuantos más bits se utilicen para representar cada muestra, mayor será la fidelidad de la señal en destino y, por tanto, menor el ruido de cuantificación. El inconveniente es que la velocidad de transmisión necesaria (ancho de banda) es directamente proporcional al número de bits de la muestra. En el caso de utilizarse 2 bits por muestra, la velocidad de transmisión necesaria sería de 16 Kbps (2 bits x 8.000 muestras al segundo).

Una forma de mejorar el ruido de cuantificación sin aumentar el ancho de banda es dejar de aplicar una cuantificación uniforme para dedicarle más valores a los niveles bajos y menos a los niveles altos. Siguiendo con el ejemplo anterior, a los valores reales 8 y 28 les corresponderían los niveles 10 y 30. Aunque, de forma absoluta, la diferencia en ambos casos es 2, esta diferencia supone un 20% de error en el primer caso y un 7% de error en el segundo. Para evitar esto se utiliza una cuantificación no uniforme (por ejemplo, logarítmica), aplicando una mayor granularidad a los niveles bajos que a los altos. En el ejemplo, a las cuatro combinaciones posibles se les aplicarían las amplitudes 0, 4, 12 y 30.

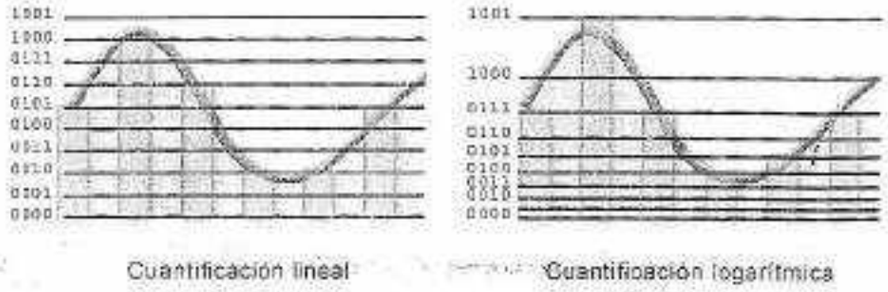

Figura 3.6. Cuantificación lineal y logarítmica

Desde 1972, las redes telefónicas han digitalizado la voz utilizando la recomendación G711 de la UIT-T. Este codec es de tipo forma de onda y utiliza 8 bits para cada muestra, lo que hace que la velocidad de transmisión de un canal telefónico sea 64 Kbps (8 x 8.000). Los equipos digitales de conmutación de circuitos y los equipos digitales de transmisión de las redes telefónicas utilizan circuitos telefónicos de 64 Kbps siguiendo esta recomendación.

El Vocoder

Homer W. Dudley era un físico que trabajaba en los laboratorios Bell Telephone. En 1928 se le ocurrió el reto de reducir el ancho de banda necesario para transmitir la voz por las redes de telefonía. De esta forma se incrementaría la capacidad de transmisión de las redes de larga distancia sin tener que crear nueva infraestructura. Desarrolló una idea que consistía en analizar la señal de la voz para identificar los parámetros que la definen, de forma que transmitiendo estos parámetros se pueda resintetizar la voz en destino. A este proceso le dio el nombre de *análisis y resíntesis del paso de banda de la voz*.

Sus teorías las llevó a la práctica con un prototipo al que llamó Vocoder (*Voice Coder* o 'Codificador de voz'). Con el tiempo, el vocoder evolucionó a un diseño más viable comercialmente y se le cambió el nombre por el de Voder. Este equipo se mostraría en la Feria Mundial de Nueva York de 1939.

El Voder nunca tuvo un éxito comercial, entre otras cosas, porque la idea de transformar las conversaciones telefónicas de la gente en murmullos robóticos no le pareció atractiva a los ejecutivos de las compañías telefónicas. No obstante, los militares descubrieron que este sistema facilitaba el cifrado de las comunicaciones, con lo que durante la segunda guerra mundial desarrollaron una versión digital (1942) que sería utilizada para establecer comunicaciones cifradas entre Roosevelt y Churchill.

ESTÁNDARES DE CODEC

Desde que Alec H. Reeves tuviera la idea de digitalizar la señal de la voz en 1937, se han ido desarrollando sistemas cada vez más eficientes y eficaces. Desde un principio, el objetivo ha sido el mismo: conseguir un sistema que ofrezca una mayor calidad de sonido con un menor ancho de banda. El tiempo y el esfuerzo de muchos investigadores y entidades han propiciado que esta marca se superara una y otra vez. Sin embargo, este esfuerzo resultaría banal si el mercado no se hubiera puesto de acuerdo en utilizar sistemas compatibles entre sí.

El organismo regulador que ha jugado el papel más destacado con los codecs de voz ha sido la Unión Internacional de Telecomunicaciones, sector de normalización de las telecomunicaciones, más conocida como UIT-T o por sus siglas en inglés ITU-T.

ESTÁNDAR UIT	PUBLICADO (Año)	CODEC UTILIZADO	VELOCIDAD (Kbps)	SEGMENTO (ms)	Segmentos/segundos	CALIDAD MOS
G.711	1988	PCM	64	8	8.000	4,3
G.721	1988	ADPCM	32	4	8.000	4,0
G.723	1988	ADPCM	24-40	3-5	8.000	
G.726	1990	ADPCM	16-40	2-5	8.000	4,0 (32Kbps)
G.727	1990	ADPCM	16-64	2-8	8.000	4,0
G.728	1992	LD-CELP	16	10	1.600	3,9
G.729	1996	CS-ACELP	8	80	100	4,0
G.729A	1996	CS-ACELP	8	80	100	3,7
G.729B	1996	CS-ACELP	8	80	100	
G.729C	1998	CS-ACELP	8	80	100	
G.729D	1998	CS-ACELP	6,4			3,8
G.729E	1998	CS-ACELP	11,8			
G.723.1	1996	ACELP	5,3	160	33,3	3,6
G.723.1	1996	MP-MLQ	6,3	192	33,3	3,8

Tabla 3.3. Comparación de formatos estándar de codec

A diferencia de las redes telefónicas tradicionales, formadas por operadores interconectados donde cada uno de ellos controla su red de forma absoluta, Internet está formada por proveedores de red interconectados sobre los que otros proveedores ofrecen servicios. Esto hace que el trabajo de regulación sea más complicado si cabe. Por ejemplo, una particularidad que se puede dar con la VoIP es que una misma comunicación de voz se vea sometida a distintas codificaciones en cascada. El resultado es que la calidad será peor que la del peor de los tramos.

Otra particularidad de los codecs es que están optimizados para reproducir el habla, sin embargo, a veces se necesita transmitir otro tipo de información distinta de la voz. Piénsese, por ejemplo, en el fax o en los tonos de control del contestador automático.

En estos casos, los proveedores de servicio suelen buscar alternativas que garanticen la calidad y no limiten, en lo posible, los servicios a los que los usuarios de las redes telefónicas tradicionales estamos acostumbrados.

Las recomendaciones de la UIT-T que se ocupan de normalizar la codificación de la voz son las de la serie G.700.

G.711. PCM

En 1937, Alec Harley Reeves, un brillante ingeniero británico que trabajaba en Francia para la International Western Electric Company desarrolló la idea de digitalizar la señal de voz para evitar los ruidos y distorsiones a las que se ven sometidas las comunicaciones analógicas. El resultado fue la técnica de codificación conocida como PCM (*Pulse Code Modulation*, 'Modulación por codificación de pulsos'). Como, por otro lado, las comunicaciones analógicas podían ser interceptadas fácilmente, las primeras aplicaciones de este sistema de audio digital fueron militares. No obstante, la codificación PCM no se haría popular hasta los años 60, después de la invención del transistor.

Curiosamente, Reeves, quien también es considerado precursor del radar, se interesó también por las comunicaciones paranormales, como la telepatía o las comunicaciones con el más allá; pero en este campo no consiguió avances.

El sistema de digitalización PCM quedaría normalizado por el UIT-T con el modelo G.711, publicado en 1972. Dado que esta norma es utilizada por las redes telefónicas para la digitalización de la voz, se puede decir que esta es la técnica de codificación más utilizada hoy en día.

PCM es un codec de forma de onda que utiliza una velocidad de muestreo de 8.000 Hz. Aunque en un principio utilizaba una cuantificación uniforme de 12 bits por muestra, lo que produce un ancho de banda de 96 Kbps, posteriormente se desarrollaron distintos tipos de cuantificaciones no uniformes que sólo necesitan 8 bits por muestra, lo que nos lleva a un ancho de banda de 64 Kbps.

G.711 utiliza una cuantificación no uniforme en dos variantes:

- **Ley μ** (Ley mu). Utilizada, fundamentalmente, en Estados Unidos y Japón.
- **Ley A**. Propuesta por K. Cattermole en 1962 y utilizada en Europa y el resto del mundo.

Lo que diferencia a estas dos variantes es la forma de realizar la cuantificación.

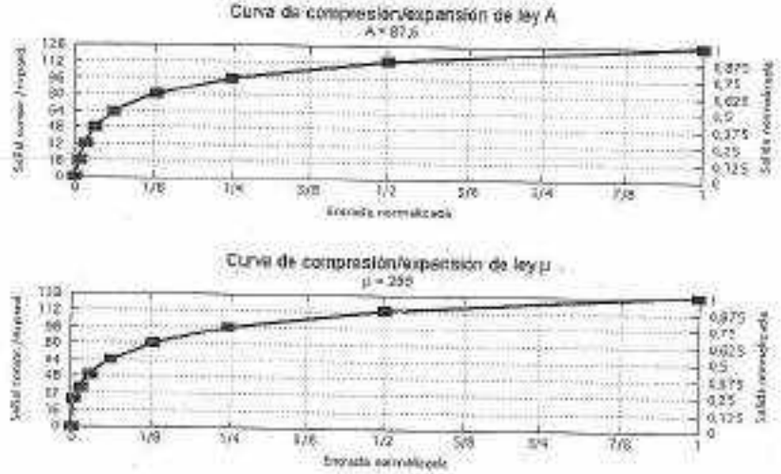

Figura 3.7. Curvas de compresión/expansión

G.726. PCM diferencial y adaptativo

El sistema PCM transmite muestras individuales para reconstruir en destino la forma de onda original. Este sistema no considera ningún tipo de relación entre muestras consecutivas. No obstante, el hecho es que la señal de la voz va evolucionando de forma suave. Esto quiere decir que si se tienen en cuenta las últimas muestras, se puede predecir, aproximadamente, el valor que va a tener la muestra siguiente. Por tanto, bastará con transmitir la diferencia entre el valor calculado y el valor real de la muestra para que en el destino puedan reconstruir la señal original. Como en el otro extremo se utiliza el mismo modelo de predicción, al recibir el valor de la diferencia se consigue reproducir la señal original. La gran ventaja de esta técnica es que la información que se necesita transmitir es mínima, reduciéndose significativamente el ancho de banda necesario (se queda en un 25% comparado con PCM).

Esta técnica se conoce como PCM diferencial o DPCM (*Differential PCM* o 'PCM diferencial').

En la idea de reducir aún más el ancho de banda necesario se pensó que los parámetros de predicción se podrían ir adaptando (modificando según los resultados que se van obteniendo) para minimizar el error de predicción. Como en ambos extremos se utiliza el mismo modelo de predicción, se consigue el mismo resultado pero con menos información que transmitir. A esta técnica se la conoce como DPCM adaptativo o ADPCM.

El sistema ADPCM está recogido en el estándar G.721 de la UIT-T. Con este estándar se consigue un canal de voz con calidad similar a G.711 utilizando tan sólo 32 Kbps. G.721 fue sustituido en 1984 por G.726, el cual utiliza canales de 16, 24, 32 y 40 Kbps con diferentes niveles de calidad.

Tanto los sistemas DPCM como ADPCM son del tipo codec de forma de onda.

G.728. LD-CELP

A principio de los años 80 empezaron a aparecer los primeros codecs híbridos. Este es el caso de los sistemas conocidos como AbS (*Analysis by Synthesis*, 'Análisis por síntesis'), los cuales se basan en utilizar una lista de modelos de señal de forma que sólo tienen que transmitir al destino el código de identificación del modelo que le va correspondiendo a cada muestra de señal.

El primer sistema AbS apareció en 1982 y se conoce como MPE (*Multi-Pulse Excited*, 'Excitación multipulso'). En 1986 aparecería una variante conocida como RPE (*Regular-Pulse Excited*, 'Excitación de pulso regular'), utilizada por el sistema europeo de telefonía móvil GSM (*Global System for Mobile communications*, 'Sistema global para las comunicaciones móviles'), consiguiendo una alta calidad de voz con tan sólo 13 Kbps. A este sistema se lo conoce como GSM-FR (*GSM Full rate*, 'GSM de velocidad completa').

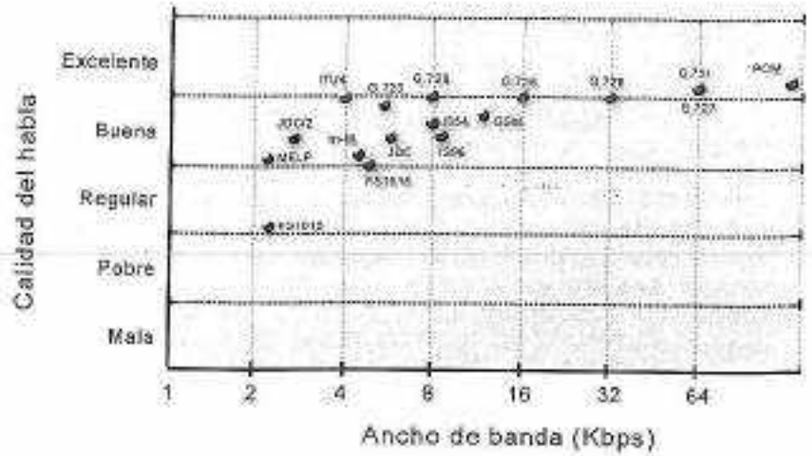

Figura 3.8. Comparación de los distintos codecs

El inconveniente de los sistemas MPE y RPE es que no pueden mantener una buena calidad de voz a velocidades inferiores a 10 Kbps. Este inconveniente fue superado por un nuevo codec conocido como CELP (*Code-Excited Linear Predictive*, 'Predicción lineal con excitación por código').

La recomendación UIT-T G.728 describe una modalidad del sistema CELP conocida como LD-CELP (*Low-Delay CELP*, 'CELP de bajo retardo'). G.728 agrupa cinco muestras de voz para determinar mejor el vector del libro de códigos que le corresponde. Este proceso de análisis genera un retardo adicional de menos de un milisegundo, de ahí el nombre.

LD-CELP utiliza un libro de códigos de 1.024 entradas, por lo que el índice del libro de códigos sólo necesita 10 bits (RPE utiliza 47 bits). Como G.728 toma 8.000 muestras por segundo y envía 10 bits por cada 5 muestras, el ancho de banda necesario es de 16 Kbps. La calidad de voz conseguida es 3,9 MOS.

El gran inconveniente del sistema G.728 es que necesita procesadores potentes que sean capaces de llevar a cabo el análisis necesario, lo que hace que raramente sea utilizado debido a su mayor coste.

G.723.1. ACELP y MP-MLQ

En la carrera por conseguir disminuir el ancho de banda necesario manteniendo unos buenos niveles de calidad, en 1995 se publicó el nuevo estándar G.723.1 que consigue velocidades de 6,3 y 5,3 Kbps con calidades de voz 3,8 y 3,5 respectivamente.

Estos codecs realizan 8.000 muestras por segundo que cuantifican con 16 bits. A continuación toman cuatro grupos de 60 muestras (3.840 bits) para determinar los coeficientes de predicción apropiados. Los parámetros que identifican el modelo del libro de códigos utilizado se envían de acuerdo a estos dos sistemas:

- MP-MLQ (*Multipulse Maximum Likelihood Quantization*, 'Cuantificación multipulso de máxima probabilidad'). Consigue un ancho de banda de 6,4 Kbps enviando la información comprimida en tramas de 192 bits (8.000 x 16 x 192 / 3.840 = 5.333).
- ACELP (*Algebraic Code-Excited Linear Prediction*, 'Predicción lineal con excitación por código algebraico'). Consigue reducir el ancho de banda necesario a 5,3 Kbps al enviar la información comprimida en tramas de 160 bits (8.000 x 16 x 160 / 3.840 = 5.333). Lo que aporta ACELP es que las entradas del libro de códigos ya no vienen dadas por un conjunto de valores que definen la forma de onda, sino que dichas formas de onda se representan por ecuaciones algebraicas.

Adicionalmente, G.723.1 utiliza un sistema de supresión del silencio conocido como SID (*Silence Insertion Description*, 'Descripción de la inserción de silencio'). Este sistema solo necesita transmitir 4 octetos, por lo que el silencio sólo ocupa 1 Kbps.

Estándar UIT	Codificación	Ancho de banda (Kbps)	Complejidad (MIPS)	Tamaño de trama (ms)	Calidad (MOS)
G.711	*PCM*	*64*	*<1*	*0,125*	*4,3*
G.721	*ADPCM*	*32*	*10*	*0,125*	*4,0*
G.723.1	*MP-MLQ/ ACELP*	*6,3/5,3*	*20-25*	*37,5*	*3,7-3,9*
G.726	*ADPCM*	*16/24/32/40*	*1,25*	*0,125*	*4,0*
G.727	*ADPCM*	*16/24/32/40*	*12*	*0,125*	*4,0*
G.728	*LD-CELP*	*16*	*33*	*0,625*	*3,9*
G.729	*CS-ACELP*	*8*	*22*	*15*	*4,0*
GSM FR	*RPE LTP*	*13*	*4,5*	*20*	*3,6-3,8*
GSM EFR	*ACELP*	*4,75-12,2*	*15-20*	*20*	*4,0*

Tabla 3.4. Técnicas de codificación de la voz

El inconveniente de los sistemas G.723.1 es el gran retardo total que genera (superior a 37,5 milisegundos), debido, entre otras cosas al tiempo necesario para llevar a cabo el análisis (7,5 milisegundos). Las calidades conseguidas con este codec están en torno a los 3,8 MOS. No obstante, en enlaces que añadan retardos excesivos los niveles de calidad bajan tremendamente. A pesar de todo, este tipo de codificación resulta muy interesante en aplicaciones que disponen de muy poco ancho de banda, como por ejemplo, en el acceso a Internet por red telefónica.

G.729. CS-ACELP

Otras de las líneas de trabajo que se han seguido para reducir el ancho de banda necesario ha sido el sistema conocido como CS-ACELP (*Conjugate Structure ACELP*, 'ACELP con estructura conjugada'). Con este sistema se consigue que el ancho de banda necesario sea de 8 Kbps. Para ello se toman 8.000 muestras por segundo y se identifican los parámetros de predicción y modelo del libro de códigos con cada grupo de 80 muestras. Al destino se envían tramas de 80 bits por cada grupo. El resultado es 8 Kbps (8.000x80/80).

CS-ACELP utiliza dos libros de códigos: uno fijo y otro adaptable. El primero de ellos contiene formas de onda preestablecidas, mientras que el segundo contiene formas de onda que se van adaptando a las señales que se van reconstruyendo. El resultado es una mayor calidad de voz. Para el cálculo de los parámetros de predicción no sólo se utiliza el grupo de 80 muestras (10 milisegundos), sino que también se tienen en cuenta las muestras de los 5 milisegundos siguientes. El retardo del algoritmo es 15 milisegundos.

El estándar G.729 de UIT-T se basa en este sistema. Posteriormente se publicarían distintas mejoras: G.729A (MOS 3,7), G.729B (que le saca mayor eficiencia a los periodos de silencio), G.729D (que trabaja a 6,4 Kbps con MOS 3,8) y G.729E (utiliza 11,8 Kbps pero ofrece una mayor calidad de voz).

Otros codificadores

Aparte de los codecs normalizados descritos anteriormente, en el mercado se han desarrollado otro tipo de codecs que han sido utilizados, fundamentalmente, para los sistemas de telefonía móvil digital. Este es el caso, por ejemplo, de los sistemas siguientes:

- CDMA QCELP (*Qualcom CELP*, IS-733). Se trata de un codificador de velocidad variable que se utiliza actualmente en los sistemas de telefonía móvil del tipo CDMA (IS-95). Aunque QCELP puede funcionar a distintas velocidades, las más comunes son 6,2 y 13,3 Kbps.
- GSM EFR (*Enhanced FR*, 'FR mejorado'). Este sistema, utilizado como alternativa al original FR se basa en la técnica ACELP, trabaja a 12.200 bps (FR lo hacía a 13.900 bps) y ofrece una mejor calidad que FR.
- GSM AMR (*Adaptative Multi-Rate*, 'Multivelocidad adaptable'). Se trata de un sistema que puede trabajar en 8 modos distintos con velocidades que van desde 4,75 a 12,2 Kbps. Cuando opera a 12,2 se trata del sistema EFR. El sistema puede cambiar de un modo a otro cada 20 milisegundos para adaptarse a las condiciones del entorno. El sistema AMR es utilizado para la telefonía móvil de tercera generación.

Internet por dentro

INTRODUCCIÓN

La mayoría de los usuarios de Internet han oído hablar del término TCP/IP. Quizás no se sepa qué es, pero sí se sabe que tiene algo que ver con el funcionamiento de Internet. Para comprender el porqué de los distintos protocolos que se utilizan en la telefonía IP es necesario tener unos conocimientos básicos sobre TCP/IP.

TCP/IP es el idioma que hablan los ordenadores cuando se comunican a través de Internet. Afortunadamente, el idioma TCP/IP es mucho más fácil de aprender que cualquier idioma humano. Además, sólo los ordenadores necesitan comprender sus matices.

TCP/IP está formado por un conjunto de normas, un conjunto de protocolos, que al utilizarse coordinadamente permiten establecer una comunicación entre dos o más equipos informáticos, intercambiar información entre ellos y dar por concluida dicha comunicación dejando libre los recursos utilizados.

TCP/IP no es el único conjunto de protocolos de comunicaciones que existe. A lo largo de la historia han existido muchos otros, cada uno ha nacido para cubrir unas necesidades concretas bajo unas determinadas circunstancias. Al cambiar las necesidades o las circunstancias los protocolos han ido evolucionando, dando paso a otros mejor adaptados. Además de TCP/IP, hoy día siguen existiendo algunos protocolos distintos, como es el caso de IPX/SPX (*Internetwork Packet Exchange/Sequenced Packet Exchange*, utilizado principalmente por Novell) o AppleTalk (utilizado principalmente por Apple para sus ordenadores Macintosh). No obstante, TCP/IP está poco a poco reemplazando a todos estos protocolos.

PROTOCOLOS DE COMUNICACIONES

En cualquier comunicación, bien sea entre personas o entre máquinas, siempre hace falta una serie de normas que regulen dicho proceso. En el caso de las comunicaciones entre personas, las normas las establece la sociedad y son aplicadas por cada persona de acuerdo con la educación que haya recibido; en el caso de las máquinas, las normas las establecen los organismos internacionales de normalización y son aplicadas por los ordenadores de acuerdo con el software que se tenga instalado.

Existen reuniones de personas en las que una de ellas es la que controla toda la comunicación, dando la palabra en cada momento a la persona que estime oportuno de acuerdo con los que hayan manifestado su intención de hablar. También existen reuniones en las que el que logra hablar es el que "primero salta". Pues bien, de manera similar, existen redes de ordenadores donde uno de ellos es el que controla toda la comunicación (sistemas centralizados), como también existen redes donde el ordenador que transmite por la red es el primero que la encuentra vacía (por ejemplo, Ethernet).

Obviamente, aunque existen similitudes de procedimientos, la diferencia fundamental entre personas y máquinas es que las personas están dotadas de inteligencia y pueden adaptarse fácilmente a situaciones imprevistas, esto es, tienen inventiva y capacidad de resolver situaciones nuevas. Los ordenadores, sin embargo, deben tener protocolos muy estrictos que tengan previsto todos los posibles casos que se puedan presentar en una comunicación, sin dejar nada al azar.

En definitiva, un protocolo no es más que un conjunto de reglas que emplean dos equipos informáticos para dialogar entre sí de forma que puedan establecer y mantener una comunicación libre de errores.

En un principio, cada fabricante establecía los procedimientos de comunicación de sus equipos, siendo casi imposible conectar equipos de fabricantes distintos. Con la expansión de la informática, se hizo evidente que era necesario disponer de protocolos normalizados que permitiesen la interconexión de equipos independientemente de su fabricante. Con esta idea, a lo largo de los años se han ido imponiendo distintos protocolos, cada uno de los cuales ha dominado en un campo de aplicación concreto. Tal es el caso de Ethernet en redes de área local cableada o Wi-Fi en redes de área local inalámbrica.

EL MODELO DE CAPAS

Los protocolos de comunicaciones se deben ocupar de definir todos y cada uno de los detalles de una comunicación. Esto incluye desde el aspecto físico de los conectores, los niveles de tensiones o las frecuencias a utilizar, hasta la forma de dividir la información en paquetes, cómo enrutar los datos o qué combinación de bits representa a la letra A.

Por otro lado, hay que tener en cuenta que en el proceso de comunicación intervienen muchos actores. En el mercado existen empresas que fabrican conectores, otras fabrican tarjetas de red, otras operan redes de comunicaciones, otras desarrollan aplicaciones o, incluso otras, ofrecen servicios finales. Lo cierto es que ninguna de estas empresas existiría sin las demás. Para que distintas empresas, distintos grupos de trabajo, puedan desarrollar su labor de forma independiente se ha definido un modelo en el cual se divide el proceso de la comunicación en partes independientes, llamadas **capas** (*layer* en inglés). Cada una de estas capas tiene definida una función específica, así como la forma de relacionarse con sus capas vecinas. Este modelo de trabajo se conoce con el nombre de **modelo de referencia OSI** o **modelo de capas**.

Otra de las ventajas de hacer una división por capas es que cada una de ellas puede ser normalizada de forma independiente. Si se desarrolla una nueva tecnología de cualquier aspecto de la comunicación, bastará con definir las normas de la capa correspondiente manteniendo intacta la interfaz con las capas vecinas.

Por ejemplo, la tecnología Wi-Fi sólo afecta a las dos primeras capas de modelo OSI, permaneciendo invariable el resto de capas. Gracias a esto un ordenador puede mantener una comunicación inalámbrica simplemente añadiéndole una tarjeta Wi-Fi, el resto de componentes de la comunicación son los mismos: se utiliza el mismo programa navegador, la conexión con el proveedor de acceso es la misma y la forma de trabajar en Internet también.

El modelo OSI

La Organización Internacional para la Normalización, ISO (*International Organization for Standardization*, aunque estas siglas también se hacen corresponder con *International Standard Organization*), propuso un modelo de comunicación llamado OSI (*Open System Interconnection*, 'Interconexión de sistemas abiertos') que está basado en la existencia de siete capas.

Las capas del modelo OSI son las siguientes:

1. **Capa física**. Esta capa se ocupa de las propiedades físicas que utilizarán los equipos de comunicaciones (niveles de tensiones, frecuencias de radio, codificación de las señales, dimensiones de los conectores, etc.).
2. **Capa de enlace**. Ésta se ocupa de organizar los datos, tanto de información como de control, para que las señales eléctricas puedan viajar de un equipo a otro de una red de una forma fiable (sin errores). Esto incluye dos funciones principales: facilitar los mecanismos para que las señales eléctricas lleguen inequívocamente al destinatario y permitir interconectar distintas redes entre sí (*bridging*). La capa de enlace proporciona el control de la capa física.
3. **Capa de red**. Esta capa organiza las cosas para que distintas comunicaciones puedan hacer uso de una infraestructura común, una red. Por ejemplo, aquí están definidos cómo se identifican los terminales (numeración) o cómo se enrutan los datos. En los protocolos de comunicaciones donde no se hace uso de una red no sería necesario definir esta capa.
4. **Capa de transporte**. Define los mecanismos para poder intercambiar datos entre los sistemas finales. Sería responsable del establecimiento y finalización de una comunicación entre dos equipos distantes interconectados por una red. Por ejemplo, es responsable de asegurar que los datos enviados se corresponden con los recibidos.
5. **Capa de sesión**. Se ocupa de agrupar los datos relacionados con una misma función. Esta capa organiza y sincroniza el intercambio de datos entre aplicaciones. Dicho de otra forma, controla el flujo de información.
6. **Capa de presentación**. Define el formato de los datos. Esta capa convierte los formatos de datos particulares (formatos de archivos, de caracteres ASCII, EBCDIC, etc.) en un formato común que es independiente del tipo de ordenador o de la aplicación utilizada. En el otro extremo se realizará la conversión inversa antes de entregar los datos al ordenador o aplicación de destino.
7. **Capa de aplicación**. Define cómo interactúan los datos con las aplicaciones específicas (correo electrónico, navegador, etc.).

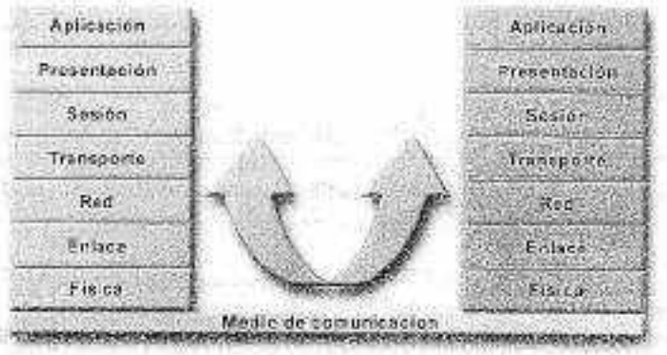

Figura 4.1. Esquema de comunicación con el modelo OSI

Desde el punto de vista práctico, lo que significa el modelo de capas es que cuando se va a transmitir una información, se divide en paquetes de una longitud determinada, y cada capa le va añadiendo a cada paquete sus propios bits de control (su propia cabecera). El resultado es que el tamaño total del paquete va creciendo hasta llegar a la capa más baja. En el destino, cada capa va eliminando su cabecera correspondiente hasta que, finalmente, le llega la información original a la capa de aplicación.

Los modelos como OSI pretenden definir todos y cada uno de los factores que intervienen en una comunicación de una red abierta; sin embargo, no todas las comunicaciones de datos son iguales; de hecho, existen comunicaciones en las que no hace falta definir una determinada capa (como en las comunicaciones directas entre dos ordenadores, en las que no es necesario que exista un nivel de red).

EL MODELO INTERNET

El modelo OSI es una referencia para todos los sistemas de comunicaciones. La gran ventaja de este modelo es que tiene un gran reconocimiento internacional y ha demostrado su eficacia como modelo desde que empezó su desarrollo en los años 80.

Por el contrario, Internet utiliza el conjunto de protocolos conocido como TCP/IP (*Transmission Control Protocol/Internet Protocol*, 'Protocolo de control de transmisión/Protocolo Internet'). Estos protocolos se desarrollaron con anterioridad a que quedase definido el modelo OSI. Por otro lado, TCP/IP representa en realidad una familia de protocolos que se ha ido definiendo anárquicamente y que a posteriori se ha estructurado en capas. De una u otra forma, el resultado es que se tiene un modelo Internet de capas que es levemente distinto al modelo OSI.

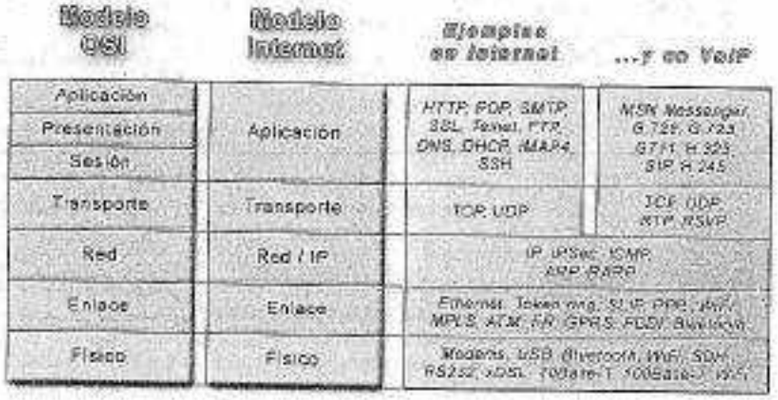

Figura 4.2. Interrelación de los protocolos usados en Internet

El modelo Internet (también conocido como modelo TCP/IP) contempla la existencia de cinco capas. Desgraciadamente, los nombres de las distintas capas no están normalizados, así que la mayoría de ellas reciben distintos nombres dependiendo de la fuente consultada:

1. **Capa física**. Equivalente a la capa física del modelo OSI.
2. **Capa de enlace**. Equivalente a la capa de enlace del modelo OSI. También se la conoce como capa MAC.
3. **Capa de red**. Equivalente a la capa de red del modelo OSI. También recibe los nombres de capa Internet, Internet o capa IP.
4. **Capa de transporte**. Equivalente a la capa de transporte del modelo OSI. También se la conoce como capa TCP.
5. **Capa aplicación**. Equivalente a las capas de sesión, presentación y aplicación del modelo OSI. El modelo TCP/IP no contempla estas tres capas como capas independientes.

Las capas física y de enlace están estrechamente unidas, por lo que en muchos casos, el mismo protocolo se encarga de definir ambas capas. Éste es el caso, por

ejemplo, del protocolo Wi-Fi (IEEE 802.11b), el cual define las capas física y de enlace que permiten crear una red local inalámbrica. De hecho, en muchos documentos se definen estas dos capas como una sola, a la que se le da el nombre de capa de acceso a red (*Network Access Layer*).

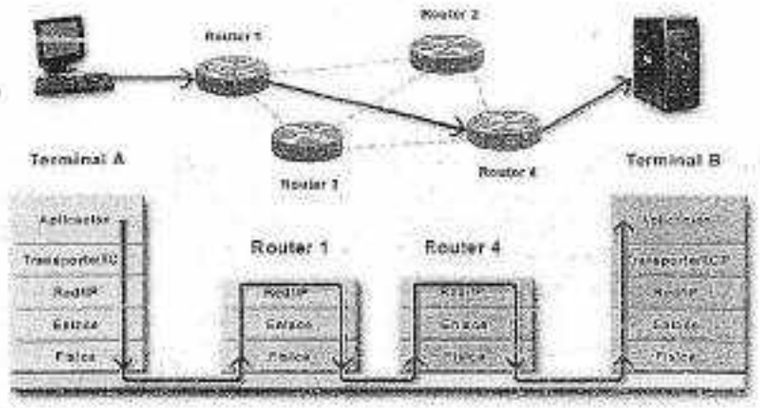

Figura 4.3. Proceso de análisis de los paquetes en una comunicación.

El conjunto de protocolos de Internet contempla más de 100 protocolos distintos. Donde existe una mayor diversidad es en la capa física y de enlace por un lado, y en la capa aplicación por otro. Por ejemplo, el acceso de un equipo a una red puede hacerse a través de cable Ethernet, Wi-Fi, llamada telefónica, etc. Cada una de estas modalidades supone la utilización de protocolos distintos al nivel físico y de enlace. Por otro lado, cada aplicación de Internet (Web, correo electrónico, transferencia de archivos, etc.) utiliza sus propios protocolos en la capa aplicación. No obstante, todas las comunicaciones de Internet coinciden en que la capa de red siempre es IP, y la capa de transporte es TCP o UDP.

Protocolos de la capa aplicación

Los protocolos de la capa aplicación definen cómo se pasan los datos de un programa a otro entre los equipos extremos de la comunicación. Por ejemplo, define cómo se transfieren los archivos, cómo se envían los mensajes de correo electrónico, cómo se transfiere la información Web, etc.

Entre la gran variedad de protocolos de este tipo, los más utilizados son los siguientes:

- **HTTP** (*Hypertext Transfer Protocol*, 'Protocolo de transferencia de hipertextos') es el protocolo de aplicación utilizado en la transferencia de páginas Web entre un servidor y un cliente Web. Este protocolo utiliza TCP.
- **SMTP** (*Simple Mail Transfer Protocol*, 'Protocolo simple de transferencia de correo') se utiliza para transmitir mensajes de correo electrónico sobre una conexión TCP. Este protocolo especifica el formato de los mensajes pero no cómo deben ser editados. Es necesario tener un editor local o una aplicación de correo electrónico para hacerlo.
- **POP3** *Post Office Protocol*, 'Protocolo de oficina de correo', versión 3) es otro protocolo de correo electrónico. Este protocolo se utiliza para que un programa cliente de correo electrónico se baje los mensajes de su buzón del servidor. Estas comunicaciones también es TCP.
- **DNS** (*Domain Name System*, 'Sistema de nombres de dominio') se utiliza para convertir nombres de dominios en direcciones IP. Los usuarios de Internet utilizamos habitualmente los nombres de dominio, mientras que la red necesita utilizar internamente los números IP correspondientes. Esta conversión de nombres de dominio a números IP la realizan los servidores DNS. Pues bien, la comunicación del software del usuario con estos servidores y de éstos entre sí se realiza mediante el protocolo DNS. Estas comunicaciones suelen ser UDP sobre el puerto 53.
- **Telnet**. Telnet permite que desde un ordenador con un programa cliente Telnet se pueda trabajar de forma remota sobre otro ordenador con una aplicación de servidor Telnet. El protocolo Telnet interconecta al programa cliente con el servidor. Los caracteres que se teclean en el cliente local son enviados por la red y procesados en el ordenador remoto. El resultado de su ejecución se muestra en la pantalla del ordenador local. Los servidores Telnet escuchan las conexiones entrantes en modo TCP en el puerto 23.
- **FTP** (*File Transfer Protocol*, 'Protocolo de transferencia de archivos') es utilizado para realizar transferencias de archivos desde o hacia un servidor FTP. Este protocolo contempla el intercambio de mensajes de control (nombre de usuario, clave, nombres de archivos y acciones a realizar) y la transferencia de archivos propiamente dicha. El archivo se transfiere sobre una conexión de datos TCP, sin la utilización de ninguna cabecera o información de control en la capa de aplicación. Cuando se completa la transferencia, se usa la conexión TCP de control para indicar que la transferencia se ha terminado y aceptar nuevos comandos. El servidor FTP escucha en el puerto 21 en TCP para las conexiones entrantes. Los clientes de FTP suelen utilizar el puerto 20.

Protocolos de transporte. TCP/UDP

El protocolo IP fija las normas para que los paquetes alcancen su destino, pero lo que no garantiza es cuándo, cuántos o en qué orden lo van a hacer. De eso se encarga TCP/UDP.

Los servicios como correo electrónico, transferencia de archivo o acceso remoto, necesitan que la información generada en un extremo de la conexión llegue al otro extremo en el orden original y sin que se haya perdido o duplicado ningún *byte*. Para estos casos, el protocolo adecuado es TCP (*Transmission Control Protocol*, 'Protocolo de control de transmisión'). Este protocolo proporciona un flujo fiable de *bytes* en los dos sentidos de la conexión. Garantiza que los bytes que salen del nodo origen sean entregados en el nodo destino de una forma fiable, en su mismo orden y sin duplicación. TCP es lo que se conoce como **protocolo orientado a la conexión**.

NÚMERO	PROTOCOLO	DESCRIPCIÓN	REFERENCIA
0	HOPOPT	*Opción de IPv6 Hop-by-Hop*	*RFC1883*
1	ICMP	*Internet Control Message*	*RFC792*
2	IGMP	*Internet Group Management Protocol (multicast)*	*RFC1112*
3	GGP	*Gateway-to-Gateway Protocol*	*RFC823*
6	TCP	*Transmission Control*	*RFC793*
17	UDP	*User Datagram Protocol*	*RFC768*
27	RDP	*Reliable Data Protocol*	*RFC908*
41	IPv6	*IPv6*	*RFC2292*
46	RSVP	*Resource Reservation Protocol*	*RFC2205*
47	GRE	*General Routing Encapsulation*	*RFC2784*
48	MHRP	*Mobile Host Routing Protocol*	
50	ESP	*Encapsulating Security Payload (IPSec)*	*RFC 2406*
51	AH	*Authentication Header (IPSec)*	*RFC 2402*
55	MOBILE	*IP Mobility*	
57	SKIP	*Simple Key Management for Internet Protocol*	
58	IPv6-ICMP	*ICMP for IPv6*	*RFC1883*
89	OSPF	*Open Shortest Path First*	*RFC1583*
92	MTP	*Multicast Transport Protocol*	*RFC 1301*
95	MICP	*Mobile Internetworking Control Protocol*	
115	L2TP	*Layer Two Tunnelling Protocol*	*RFC 2661*
121	SMP	*Simple Message Protocol*	
123	PTP	*Performance Transparency Protocol*	
138-252	*Sin asignar*		
253-254	*Utilizado para experimentación y pruebas*		*RFC3692*
255	*Reservado*		

Tabla 4.1. Ejemplos de números de protocolos asignados por IANA

Cuando se necesita transmitir voz o vídeo, es más importante transmitir con velocidad que garantizar el hecho de que lleguen absolutamente todos los bytes, o de que no haya ningún byte duplicado. Para estos casos, el protocolo de transporte utilizado es UDP (*User Datagram Protocol*, 'Protocolo de datagrama de usuario'). UDP no tiene mecanismos para confirmar la recepción de los paquetes, por lo que, entre los paquetes enviados con este protocolo, algunos pueden llegar desordenados, duplicados o puede que no lleguen. UDP es un protocolo no orientado a la conexión.

El protocolo UDP puede ser especialmente útil en aquellas redes que ya son suficientemente fiables por sí mismas, para aquellas comunicaciones que no requieren más de un paquete, para aquellos servicios que disponen de sus propios procedimientos de corrección de error o para servicios de transmisión de voz o vídeo donde la velocidad es más importante que la fiabilidad. Hay que tener en cuenta que un paquete UDP puede contener la información necesaria para reproducir entre 10 y 40 milisegundos de sonido, por lo que la pérdida de un paquete, aun no siendo deseable, no influye grandemente en la calidad de sonido en destino, y mucho menos en el entendimiento de una conversación. De hecho, la pérdida de hasta un 5% de los paquetes puede considerarse aceptable. Otra cosa distinta es que se pierdan grupos de paquetes de forma constante.

Otro problema distinto es si los paquetes llegan desordenados. Como los paquetes UDP no tienen el concepto del orden de salida, los paquetes son entregados a la capa superior en el mismo orden en el que van llegando, lo que podría producir una pérdida de calidad. No obstante, en la práctica, los paquetes de una misma sesión suelen utilizar la misma ruta, lo que hace que lleguen en el mismo orden. La probabilidad de que esto no sea así es extremadamente pequeña.

Aunque TCP y UDP son los protocolos de la capa transporte más utilizados por las aplicaciones de Internet, en realidad existe una larga lista de protocolos de este tipo. La IANA (*Internet Assigned Numbers Authority*, 'Agencia de asignación de números de Internet') dispone de una lista completa que actualmente contiene 134 protocolos (*www.iana.org/assignments/protocol-numbers*).

Protocolo de red. IP

El protocolo IP (*Internet Protocol*, 'Protocolo Internet') determina los procedimientos de direccionamiento y encaminamiento que deben seguir todos los datos transmitidos, independientemente de la red física por la que se establezca la conexión (Ethernet, red inalámbrica, circuitos alquilados, etc.).

Para poder identificar los paquetes, el protocolo IP establece que cada equipo conectado a Internet debe tener un número de identificación conocido como número IP. Todos los paquetes incluyen en su cabecera el número IP del equipo origen y destino del mismo. Adicionalmente, el paquete incluye alguna otra información de control para facilitar el enrutamiento y reensamblado en destino.

Aparte de los paquetes que transportan información, el protocolo IP utiliza también otro tipo de paquetes de control de la comunicación. Los paquetes que transportan información proceden de la capa de transporte, mientras que los que transportan datos de control son creados por la capa IP de cualquier equipo intermediario y se quedan en la capa IP de cualquier otro equipo intermediario, incluido el ordenador de destino.

Entre los paquetes de control se encuentran los conocidos como paquetes ICMP (*Internet Control Message Protocol*, 'Protocolo de mensajes de control de Internet'), utilizados para informar de posibles problemas en la comunicación y cómo resolverlos, y los paquetes ARP (*Address Resolution Protocol*, 'Protocolo de aclaración de direcciones') y RARP (*Reverse ARP*, 'ARP inverso'), utilizados para resolver determinadas cuestiones de direccionamiento.

PAQUETES Y CABECERAS

Cada capa del modelo de comunicaciones tiene sus propias responsabilidades y, para llevarlas a cabo, necesita añadir unos ciertos datos de control. Estos datos de control son añadidos en el origen por cada capa antes de pasarle el paquete a la capa inferior. Cuando llega la información al destino, cada capa identifica sus datos de control correspondientes, los interpreta, actúa en consecuencia y, si procede, elimina esos datos de control antes de pasar el paquete a la capa superior.

En un intercambio normal de información, cada capa en el destino recibe de la capa inferior los mismos datos que dicha capa le pasó en el origen. De manera efectiva es como si los datos que salen de una capa en el origen le llegasen directamente a esa misma capa en el destino. Por tanto, los datos que salen de la aplicación del origen le llegan directamente a la aplicación del destino, lo que supone el propósito de la comunicación.

Como se ha visto anteriormente, para que la comunicación pueda estar controlada en todo momento, el flujo de información se divide en trozos pequeños, llamados paquetes, que son enviados y gestionados independientemente. Cada paquete puede tener un tamaño entre 576 y 65.535 bytes.

La gran ventaja de este sistema es que, en el caso de detectarse un error, el paquete afectado se vuelve a enviar y se sigue con la comunicación. Si no se dividiera la información en pequeños paquetes, en el caso de producirse un error, habría que empezar de nuevo desde el principio, con la pérdida de tiempo que esto supone.

Los paquetes tienen una estructura que se compone de dos partes:

- **Cabecera** (*header* en inglés). Contiene los datos de control.
- **Cuerpo** (*body* en inglés). Contiene la información.

Para hacernos una idea, uno de los datos que contiene la cabecera es la dirección IP del destinatario. Esto le permite a cada paquete alcanzar su destino. Por otro lado, cada paquete puede seguir un camino distinto, por lo que podrían llegar en distinto orden. Para garantizar el reensamblado en el orden correcto, también se incluye en la cabecera un número secuencial de ensamblado.

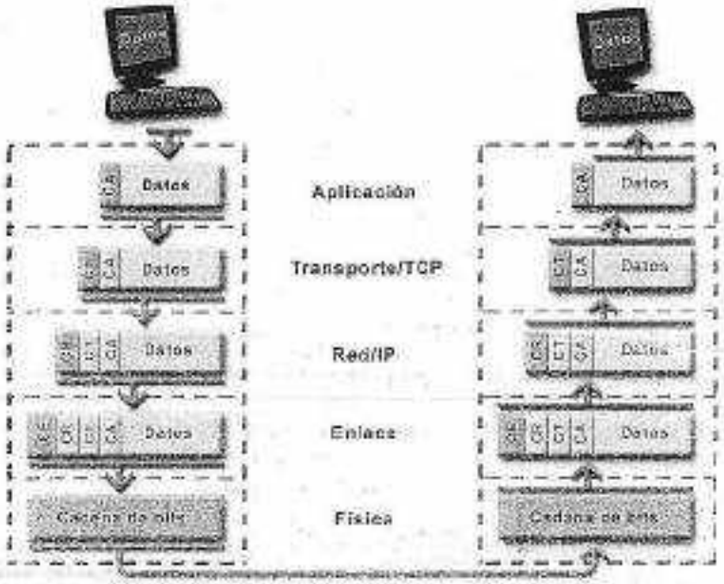

Figura 4.4. Desgajo de las cabeceras de los paquetes a su paso por las capas

Si nos fijamos, este modelo de cabecera y cuerpo no es nuevo. El envío postal tradicional utiliza un sistema similar. Se escribe una carta, se introduce en un sobre y se escribe sobre el mismo los datos de la dirección de destino, incluyendo, calle, número, ciudad, código postal, etc. El sobre constituye la cabecera, mientras que la carta que va en su interior es el cuerpo. Por otro lado, la empresa de correos maneja los datos del sobre como si se tratara de un modelo de capas. Primero se analiza el código postal para saber a qué zona enviar la carta (capa *código postal*). En el análisis del código postal no se mira el resto de datos que conforman la dirección. Al llegar la carta al distrito postal correspondiente se mira la calle para saber qué cartero hará el reparto (capa *calle*). Por último, el cartero comprueba el resto de datos de la dirección para entregar la carta a su destinatario (capa *destinatario*). Por cierto, el cartero ya no se fija en el código postal, ese dato de cabecera es de la capa código postal.

Los paquetes IP

Las cabeceras de los paquetes IP contienen todo un conjunto de datos necesarios para garantizar que el paquete se mueva rápidamente por Internet hasta alcanzar su destino y se integra con el resto de paquetes para formar la información original.

La cabecera de los paquetes IP (IPv4) está formada por los siguientes datos:

- **Versión**. Se trata de 4 bits que indican el número de la versión del protocolo IP utilizado.
- **IHL** (*Internet Header Length*, 'Longitud de la cabecera'). Indica la longitud de la cabecera en bloques de 32 bits. Generalmente, el valor es 5 (160 bits), aunque puede ser mayor si se añaden determinadas opciones al final de la misma.
- **TOS** (*Type of Service*, 'Tipo de servicio'). Se trata de ocho bits que se utilizan para fijar prioridades de confiabilidad y velocidad utilizadas por algunos servicios. Para la voz es más importante realizar la entrega de forma rápida que precisa, mientras que para la transferencia de archivos es más importante que no haya errores que la velocidad.
- **Longitud total**. Indica la longitud total del paquete.
- **Identificación**. Se trata de un número de identificación utilizado para poder reensamblar los paquetes en el orden original.
- **Indicadores** (*Flags*). Se trata de tres bits utilizados por los *routers* que manejan los paquetes:
  - El primer bit no es utilizado y está siempre a cero.
  - El segundo bit se utiliza para indicar que el paquete ha sido fragmentado. Un paquete puede tener una longitud mayor que la que pueden manejar determinados *routers*. Para resolver esto, el *router* fragmenta el paquete en partes más pequeñas y pone este bit a uno para indicar que el paquete está fragmentado. Generalmente, los fragmentos son de 576 bytes, el mínimo permitido, lo que evita que un paquete sea fragmentado dos veces.
  - El tercer bit indica que el paquete no debe fragmentarse porque el destino no es capaz de reensamblarlo. Si fuese necesaria una fragmentación y este bit estuviera a uno, simplemente se eliminaría el paquete y se informaría al remitente.
- **TTL** (*Time to Live*, 'Tiempo de vida'). Es un valor de 8 bits que se decrementa, generalmente en una unidad, cada vez que el paquete pasa por un *router*. El *router* que hace que este valor llegue a cero desecha el paquete (lo elimina) e informa al remitente. Por tanto, un paquete puede atravesar un máximo de 255 *routers* intermedios. El objetivo de este valor es impedir que los paquetes se queden perdidos indefinidamente dando vueltas por la red.
- **Protocolo**. Se trata de un valor de 8 bits que indica el protocolo de la capa transporte o aplicación que ha creado el paquete. TCP es uno de estos protocolos. El documento RFC1700 (*www.faqs.org/rfcs /rfc1700.html*) contiene un listado de estos protocolos. No obstante, esta lista no está actualizada. Puede encontrar un listado más actualizado en *www.iana.org/assignments/port-numbers*.
- **Suma de verificación** (*Checksum*). Este valor se utiliza para comprobar la integridad de la cabecera. Se asegura de que los ceros y unos que la componen son los mismos que salieron del origen.
- **IP remitente**. Se trata de la dirección IP del equipo que envía el paquete.

- **IP destino**. Se trata de la dirección IP del equipo de destino del paquete.
- **Opciones**. El campo de opciones sólo se utiliza en determinados casos, y puede contener información adicional necesaria para la comunicación. Por ejemplo, el Departamento de Defensa de Estados Unidos lo utiliza para indicar una clasificación de seguridad. También puede utilizarse para especificar la ruta a seguir por el paquete, o para registrar la ruta seguida por el mismo. Generalmente no se suele utilizar este campo.

Figura 4.5. Formato de la cabecera de un paquete IPv4.

Los paquetes TCP y UDP

Los componentes de la cabecera del protocolo TCP son los siguientes:

- **Puerto origen**. Son 16 bits que representan el número de puerto del equipo remitente. Este número identifica la aplicación concreta que transmite los datos.
- **Puerto de destino**. Se trata de 16 bits que representan el número de puerto del equipo destino. Este número identifica la aplicación concreta del equipo de destino que recibirá los datos. Existen números de puertos ya asignados a ciertas aplicaciones conocidas. Este es el caso del puerto 80, asignado al programa servidor de páginas Web.
- **Número de secuencia** (*Sequence number*). Es un número secuencial de 32 bits dedicado a asegurar que los paquetes en el destino son reensamblados en el orden correcto.
- **Confirmación** (*Acknowledgement*). El extremo que envía el paquete le indica al otro extremo que ha recibido correctamente todos los paquetes anteriores a este número.
- **Tamaño** (*Offset*). Indica la cantidad de bloques de 32 bits incluidos en la cabecera del paquete.

Figura 4.6. Formato de la cabecera de un paquete TCP

- **Reservado**. Se trata de 6 bits que están siempre puestos a cero.
- **Indicadores** (*Flags*). Se trata de 6 bits dedicados a determinadas labores de control. El objetivo de cada uno de ellos es el siguiente:
  - **URG**. Urgencia. Cuando está a uno le indica al receptor que lea el dato del campo *Puntero de urgencia*.
  - **ACK**. Cuando está a uno le indica al receptor que este paquete incluye un número de confirmación.
  - **PSH**. Envío forzado (*Push*). Cuando está a uno le indica a la capa TCP del destino que debe pasar a la capa aplicación estos datos sin demora. Esto evita que los datos puedan estar esperando en TCP a que se complete un segmento. En ocasiones, una aplicación necesita estar segura de que la aplicación de destino ha recibido todos los datos enviados.
  - **RST**. Inicializar (*Reset*). Indica que se corte la conexión actual.
  - **SYN**. Sincronización. Este bit se pone a uno en el primer paquete de una sesión TCP para indicarle al destino que empiece a contar la secuencia de números de los paquetes. Este bit nunca estará a uno durante el resto de la sesión.
  - **FIN**. Indica que el remitente ha terminado de enviar datos. Esto hace que se termine la sesión.
- **Ventana** (*Window*). Indica el número de octetos (bytes) que puede aceptar el destinatario. Esta información la envía el destinatario con cada paquete de confirmación (*Acknowledgement*). El emisor no puede enviar más datos, octetos, que los indicados por el tamaño de la ventana.
- **Suma de verificación** (*Checksum*). Este valor se utiliza para comprobar la integridad de la cabecera y de los datos. La suma de verificación de los paquetes IP verifica la integridad de la cabecera, pero la de los paquetes TCP comprueba también la integridad de los datos.
- **Puntero de datos urgentes**. Cuando el indicador URG está a uno, indica cuál es el último byte de datos que es urgente. Al destinatario le sirve para saber cuántos

datos urgentes llegan. Este campo lo utilizan algunas aplicaciones como Telnet o FTP.

- **Opciones**. Este espacio hace posible que determinadas aplicaciones puedan intercambiarse parámetros adicionales.
- **Relleno** (*Padding*). Se trata de bits de relleno para completar los 32 bits no cubiertos por los datos de opciones.

En relación con el valor del campo *Ventana*, Windows 95, 98, Me y XP fijan automáticamente el tamaño de este campo en 8 KB. Windows NT y Windows 2000 lo fijan en 16 KB. Estos tamaños están optimizados para los módems de 56 Kbps. Realmente, el tamaño adecuado de este valor depende del tiempo que necesita un paquete para viajar de un extremo a otro, latencia. Tiempos menores a 100 milisegundos se considera una latencia baja, mientras que tiempos mayores a 200 milisegundos se considera una latencia alta. Los tiempos normales están entre ambos valores.

Figura 4.7. Formato de la cabecera de un paquete UDP

A mayor tamaño de ventana menor latencia. En Windows se puede modificar el tiempo de latencia con la utilidad *Regedit*.

Para comprobar la latencia de nuestra conexión se puede utilizar la utilidad *Tracert*, presente en la mayoría de los sistemas operativos.

En cuanto a los paquetes UDP, este protocolo es muy ligero, su cabecera sólo necesita de los números de puerto origen, puerto destino, longitud del paquete UDP y suma de verificación.

Los paquetes de control ICMP

Los paquetes ICMP se utilizan para informar y resolver los posibles problemas en la comunicación que pueden tener los distintos equipos que intervienen en la misma. Por ejemplo, un equipo puede utilizar un paquete ICMP para indicarle a otro que no puede gestionar todo el tráfico que está recibiendo y que vaya más lentamente. Los paquetes ICMP pueden ser utilizados tanto por los equipos extremos de la comunicación, origen y destino, como por cualquier otro equipo intermedio.

Figura 4.8. Formato de los paquetes ICMP

Los paquetes ICMP tienen la estructura de un paquete IP donde se incluye el tipo de mensaje ICMP de que se trata, la cabecera del paquete IP original y los primeros 64 bytes de datos. Existe todo un conjunto de mensajes ICMP identificados por dos campos: tipo de mensaje y código. En su mayoría, los mensajes ICMP están definidos en la norma RFC792. Algunos de estos mensajes son los siguientes:

- **Red inaccesible** (tipo 3, código 0). Este mensaje se envía cuando un *router* no sabe por dónde tiene que enviar el paquete, ya que no identifica la dirección de destino del paquete original en sus tablas de enrutamiento.
- **Destino inaccesible** (tipo 3, código 7). Este mensaje lo envía el último *router* del enlace cuando el ordenador de destino no responde.
- **Protocolo inaccesible** (tipo 3, código 2). Este mensaje lo envía el ordenador de destino para indicar que no tiene activo el protocolo de nivel de transporte especificado.
- **Puerto inaccesible** (tipo 3, código 3). Este mensaje lo envía el ordenador de destino para indicar que no tiene ninguna aplicación asociada al puerto especificado.
- **TTL excedido** (tipo 11, código 0). Este mensaje lo envía un *router* para indicar que el decremento del campo TTL ha llegado a cero. TTL (*Time To Live*, 'Tiempo de vida') es una variable que se decrementa en una unidad cada vez que el paquete pasa por un *router*. Cuando llega a cero se desecha el paquete. Esta variable se utiliza para evitar que un paquete esté indefinidamente dando vueltas por la red. El valor inicial de TTL puede variar, pero 32 o 128 son valores habituales. En el caso de Windows, este valor está definido por la variable *DefaultTTL* de *Rededit*.
- **Petición de eco** (tipo 8, código 0). Se trata de una pregunta que se utiliza para comprobar si un equipo está activo. Si el equipo devuelve el mensaje *Respuesta de eco*, es que está activo; en caso contrario no se recibirá ninguna respuesta.
- **Respuesta de eco** (tipo 0, código 0). Se utiliza como respuesta a una petición de eco para confirmar que el equipo está activo.
- **Redireccionamiento** (tipo 5, código 0). Este mensaje lo utiliza un *router* intermedio para indicar que existe una ruta más directa para alcanzar el destino. Quien recibe este mensaje dejará de enviar los paquetes a través de dicho *router* intermedio y lo hará por el *router* alternativo indicado.

TIPO	MENSAJE	RFC
0	*Respuesta de eco*	[RFC792]
3	*Destino inalcanzable*	[RFC792]
4	*Flujo de origen*	[RFC792]
5	*Redireccionamiento*	[RFC792]
6	*Dirección alternativa*	[JBP]
8	*Solicitud de eco*	[RFC792]
9	*Anuncio del router*	[RFC1256]
10	*Petición del router*	[RFC1256]
11	*Tiempo excedido*	[RFC792]
12	*Parámetros erróneos*	[RFC792]
13	*Marca temporal*	[RFC792]
14	*Respuesta a marca temporal*	[RFC792]
15	*Petición de información*	[RFC792]
16	*Respuesta de información*	[RFC792]
17	*Petición de máscara*	[RFC950]
18	*Respuesta de máscara*	[RFC950]
30	*Traceroute*	[RFC1393]
31	*Error de conversión*	[RFC1475]
37	*Petición de nombre de dominio*	[RFC1788]
38	*Respuesta de nombre de dominio*	[RFC1788]
Más información: *www.iana.org/assignments/icmp-parameters*		

Tabla 4.2. Ejemplos de tipos de mensajes ICMP

- **Error de formato** (tipo 12). Indica que se encontró un problema durante el procesamiento de los parámetros de la cabecera IP. El paquete original se desecha y se informa al destinatario del tipo de error encontrado.
- **Flujo de origen** (tipo 4, código 0). Indica que los paquetes están llegando demasiado rápido y no pueden ser procesados.

El número MAC

Uno de los temas de los que se ocupa la capa de enlace es el de definir los procedimientos necesarios para que los datos lleguen físicamente al equipo de destino. Para ello se utiliza un método de direccionamiento conocido como MAC (*Media Access Control*, 'Control de acceso al medio'). Este método permite que distintos equipos puedan utilizar un mismo medio de comunicación (un mismo cable en el caso de redes Ethernet, o unas mismas frecuencias en el caso de redes Wi-Fi) sin interferirse unos a otros.

Los fabricantes de equipos, de tarjetas de red, identifican a cada uno de ellos con un número irrepetible. Este número se conoce como número MAC y está formado por 12 caracteres alfanuméricos (por ejemplo, 12-AB-56-78-90-FE). No existen dos tarjetas de

red, independientemente de quién sea su fabricante, que tengan el mismo número MAC. Es algo así como un código de barras individual de cada tarjeta.

El método de direccionamiento MAC identifica a cada ordenador, a cada equipo, por el número MAC de su tarjeta de red. Esto permite identificar a cada componente de una red de forma completamente inequívoca.

Por cierto, una tarjeta de red es el componente electrónico de un equipo (un ordenador, una PDA, etc.) que se ocupa de las comunicaciones. Una tarjeta de red Ethernet o un adaptador de red Wi-Fi son ejemplos de tarjetas de red. A estas tarjetas también se las conoce por su nombre en inglés: NIC (*Network Interface Card*, 'Tarjeta interfaz de red').

Los paquetes ARP

En Internet se utiliza el sistema de direccionamiento IP independientemente del destino del paquete. No obstante, determinadas redes requieren, además de la dirección IP, la dirección MAC que identifica físicamente al equipo destinatario.

Generalmente, el remitente conoce la dirección IP, pero no conoce la dirección MAC. Pues bien, los paquetes ARP (*Address Resolution Protocol*, 'Protocolo de aclaración de direcciones') se utilizan para preguntar al destinatario de una dirección IP cuál es su dirección MAC. De la misma forma, los paquetes RARP (*Reverse ARP*, 'ARP inverso') se utilizan para preguntarle al destinatario de una dirección MAC por su dirección IP. El destinatario envía un nuevo paquete ARP o RARP con la respuesta.

ESTABLECIMIENTO DE UNA SESIÓN TCP

Una sesión TCP se inicia con el envío de un paquete TCP con el bit SYN puesto a uno y un número de secuencia inicial. El extremo B responde con un paquete con los bits SYN y ACK puestos a uno, su propio número de secuencia, y el número de confirmación igual al número de secuencia enviado por el extremo A más uno.

Cuando el extremo A recibe este paquete considera que la sesión se ha establecido y responde con un paquete con el bit ACK a uno y el número de confirmación igual al número de secuencia enviado por el extremo B más uno. Con esto, el extremo B considera también que la sesión está establecida. A partir de este momento los datos pueden fluir en cualquier dirección. A este procedimiento de establecimiento de sesión se lo conoce como establecimiento en tres pasos (*Three way handshake*).

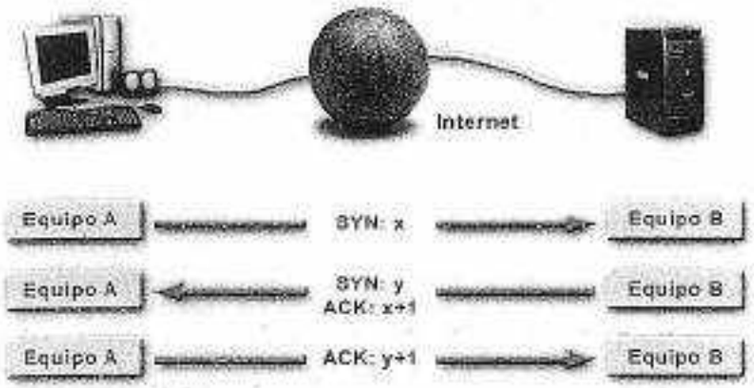

Figura 4.9. Establecimiento de sesión en tres pasos

Además de los datos anteriores, el extremo A debe incluir en su primer paquete un número de puerto de destino y de origen. El número de puerto de destino depende del tipo de servidor a que se conecte. Por ejemplo, para un servidor Web el número de puerto sería el 80 y para un servidor FTP el 20. Por su parte, el número de puerto origen dependerá de la aplicación utilizada. En cualquier caso, se tratará de un número de puerto entre 1.024 y 65.536.

En el flujo de la comunicación se puede dar la circunstancia de que un equipo envíe datos más rápidamente de lo que el otro equipo puede asimilar. Cada equipo va guardando los datos que va recibiendo del otro extremo en una memoria intermedia llamada ventana. Cuando lee los datos de la ventana la vacía y envía un mensaje de confirmación al remitente. Por tanto, en el peor de los casos (que se quede bloqueado), un equipo siempre podrá recibir, al menos, los bytes que le quepan en la ventana. Esto quiere decir que, entre cada mensaje de confirmación, cada extremo sólo debe enviar, como máximo, el número de bytes de la ventana del otro extremo. El campo ventana está incluido en todos los paquetes TCP, por lo que cada extremo de la comunicación es consciente de las limitaciones del otro extremo.

MULTIDIFUSIÓN. IP *MULTICAST*

Existen aplicaciones, como las multiconferencias, en las que el flujo de voz de un participante debe distribuirse simultáneamente a múltiples destinatarios. Evidentemente, esto puede hacerse enviando tantas copias de paquetes IP como destinatarios haya, pero esto inunda innecesariamente la red de copias de un mismo paquete. Una solución más adecuada consiste en enviar un único paquete con una dirección IP de destino que está asociada a todos los destinatarios. Esto se conoce como *multicast* o multidifusión.

Para que esto funcione es necesario definir un grupo *multicast* donde cada uno de sus miembros recibe los datagramas enviados a la dirección *multicast* del grupo. Estos miembros seguirán recibiendo los paquetes dirigidos a su propia dirección.

IPv4 reserva las direcciones 224.0.0.0 a 239.255.255.255 para direcciones *multicast*. Por tanto, los *routers* saben que estas direcciones se deben manejar de una forma particular. Algunas de las direcciones de este rango tienen un propósito especial. Éste es el caso de las siguientes:

- 224.0.0.1 hace referencia a todos los equipos (*hosts*) de la subred local.
- 224.0.0.2 hace referencia a todos los *routers* de la subred local.
- 224.0.0.5 hace referencia a todos los *router* que admitan OSPF.
- 224.0.0.9 hace referencia a todos los *routers* que admitan RIP (*Routing Information Protocol*) versión 2.

Por ejemplo, supongamos que un equipo conectado a un *router* A envía un paquete *multicast* dirigido a dos equipos conectados a un *router* C y tres equipos conectados a un *router* D. Si entre el *router* A y estos últimos existiera un *router* B, del *router* A saldría una sola copia del paquete con destino al *router* B. Este *router* enviaría una copia al *router* C y otra al D. Por último, los *routers* C y D enviarían una copia a cada equipo del grupo *multicast*. Como se puede ver, el objetivo es que el número de copias de los datagramas *multicast* sea siempre el mínimo, o lo que es lo mismo, que por una ruta no viajen más de una copia de un mismo paquete.

Para que funcione el sistema *multicast* deben cumplirse ciertas premisas: debe haber disponibilidad de direcciones de *multicast*, los equipos que se quieran unir a un grupo *multicast* particular deben informar a su *router* local, los *routers* deben poder registrar y reconocer las direcciones *multicast* asociadas a cada uno de sus equipos, así como reenviar los datagramas convenientemente. Por último, los protocolos de enrutado, como OSPF, deben permitir la propagación de la información de enrutado para direcciones *multicast*.

Para que los equipos de un *router* puedan comunicarle su pertenencia a un grupo *multicast* concreto, así como para que los distintos *routers* se intercambien este tipo de información, se utiliza el protocolo IGMP (*Internet Group Management Protocol*, 'Protocolo de gestión de grupos de Internet'). Las versiones 1, 2 y 3 de IGMP están recogidas en RFC 1112, RFC 2236 y RFC 3376, respectivamente. Esta última, publicada en 2002, incluye la posibilidad de filtrar por el origen. Esto quiere decir que un equipo puede indicar que sólo quiere recibir los paquetes *multicast* enviados por un determinado origen, o todos menos los de un origen determinado.

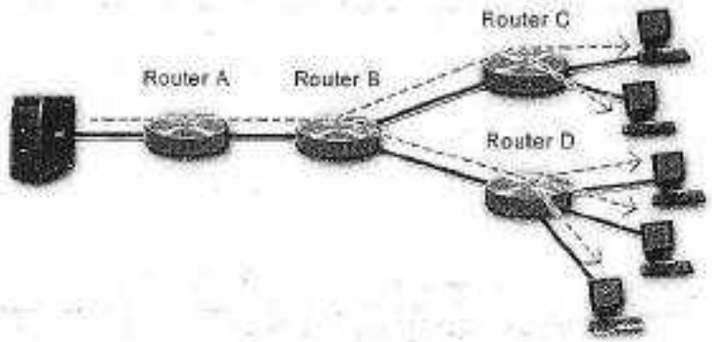

Figura 4.10. Envío en multidifusión

IPV6

Desde principios de los años 70 el sistema de direccionamiento que ha utilizado Internet ha sido el denominado IPv4. Como hemos visto, este sistema utiliza 32 bits para identificar cada una de las direcciones de los equipos conectados a Internet. Esto da capacidad para casi 4.300 millones de direcciones (4.294.967.296 direcciones exactamente). Sin embargo, los creadores de IPv4 no imaginaron el gran éxito que iba a tener este protocolo IP en tan poco tiempo. El hecho es que esta capacidad ya resulta muy escasa para las necesidades actuales de la red. En Norteamérica no se nota tanto este problema, pero en Japón o Europa el problema es creciente debido al desarrollo de las redes de telefonía celular y de tecnologías xDSL. El problema se vuelve grave en muchos países europeos del este, asiáticos o africanos que sólo tienen una clase C (255 direcciones) para todo el país.

En poco tiempo se está pasando de utilizar una dirección IP por cada diez usuarios a tener una relación 1:1. Incluso con la introducción de dispositivos 'siempre conectados' se espera que esta tendencia llegue en un futuro no muy lejano a 50:1 o 100:1.

En 1997 el número de aplicaciones con necesidades de conectarse directamente a Internet (WebTV, agendas electrónicas, juegos o telefonía sobre IP sin tener en cuenta los ordenadores) era de 3 millones. Para el año 2002 la cifra superaba los 56 millones. Como ejemplos de algunas de estas aplicaciones tenemos los siguientes:

- Terminales de telefonía de voz sobre IP (VoIP).
- Televisión y radio sobre Internet.
- Sistemas de seguridad, alarmas, incendios, televigilancia y control.
- Dispositivos domésticos (frigoríficos, lavadoras, despertadores, etc.) conectados a Internet para conseguir un mejor control o la introducción de nuevos servicios.

- Lectura de contadores del consumo de agua, gas, electricidad, etc.
- Walkman MP3 que recuperan y almacenan automáticamente las nuevas creaciones de nuestro interés.
- Nuevas aplicaciones que utilicen las nuevas tecnologías móviles e inalámbricas WAP, *bluetooth*, Wi-Fi, etc.

Podríamos hablar de que casi cualquier dispositivo doméstico o industrial podría estar integrado en la red.

IPv6 viene a resolver el problema de direccionamiento, ya que aporta 128 bits de direccionamiento. Esto da capacidad para tener 340.282.366.920.938.463.463.374.607. 431.768.211.456 direcciones. Una cifra impronunciable.

Versión (4b)
Clase de de tráfico (8 bits)
Etiqueta de flujo (20 bits)
Longitud de la carga útil (16 bits)
Siguiente cabecera (8 bits)
Límite de saltos (8 bits)
Dirección del remitente (128 bits)
Dirección del destinatario (128 bits)

Figura 4.11. Formato de la cabecera de un paquete IPv6

Aparte de resolver el problema de la falta de direcciones, IPv6 aporta los siguientes valores:

- Seguridad intrínseca en el núcleo del protocolo (IPSec).
- Facilidades de Calidad de servicio (QoS) y Clase de servicio (CoS).
- *Multicast*: envío de un mismo paquete a un grupo de receptores.
- *Anycast*: envío de un paquete a un receptor dentro de un grupo.
- Paquetes IP eficientes y extensibles, sin fragmentación de los paquetes en los *routers*, con cabecera de longitud fija y más simple.
- Posibilidad de enviar paquetes con una carga útil (información real) de más de 65.535 bytes.
- Permite un enrutamiento más eficiente gracias a disponer de una jerarquía de direccionamiento basada en la agregación.

- Renumeración y *multihoming*. *Multihoming* permite que una red tenga varios accesos diversificados a Internet.
- Características de movilidad. Esto permite facilitar servicios sin limitaciones a los terminales móviles.

A pesar de todas las ventajas de IPv6, el esfuerzo técnico y económico para migrar de un sistema a otro es bastante importante. Esto hace que se haya empezado por crear islas IPv6 que progresivamente vayan reemplazando a IPv4 en toda la red Internet.

Para más información sobre IPv6, puede consultar las siguientes direcciones:

www.ipv6forum.com *www.ipv6-taskforce.org*
www.ipv6.unam.mx *www.consulintel.es*

Direcciones IPv6

Las direcciones IPv6 están definidas en RFC 2373 y están formadas por 8 números hexadecimales de 4 cifras. En la representación de estas direcciones se separan los 8 números con el símbolo de dos puntos (por ejemplo, 54:0:0:0:0:1:8:FF). Si varios números contiguos tienen el valor cero, se pueden sustituir todos ellos por el símbolo "::". No obstante, una dirección IPv6 sólo puede contener un símbolo de este tipo. Por ejemplo, la dirección 54:0:0:0:0:1:8:FF puede representarse como 54::1:8:FF.

Por otro lado, IPv6 considera una serie de direcciones especiales:

- **Dirección con todos ceros** (::). Significa una dirección no especificada. Esta dirección puede utilizarse, por ejemplo, cuando un nodo no conoce todavía su dirección. Evidentemente, esta dirección no se debe utilizar como dirección de destino.
- **Dirección de autoenvío o *loopback*** (::1). Esta dirección la utiliza un nodo para enviarse un paquete a sí mismo.
- **Dirección compatible IPv4 tipo 1**. En este caso, los primeros 96 bits son ceros y los últimos 32 contienen la dirección IPv4. Esta representación la utilizan equipos IPv6 que van a transmitir sus paquetes por una infraestructura IPv4. La representación de esta dirección sería ::x.x.x.x.
- **Dirección compatible IPv4 tipo 2**. Se trata de una dirección IPv6 donde los primeros 80 bits son ceros, los siguientes 16 son F y los últimos 32 representan la dirección IPv4. La representación de esta dirección sería ::FFFF:x.x.x.x.

NOMBRE	DOCUMENTO
BGP (*Border Gateway Protocol*)	RFC1267, RFC1467
DNS (*Domain Name System*)	RFC1119
FTP (*File Transfer Protocol*)	RFC359
ICMP (*Internet Control Message Protocol*)	RFC792, RFC1122
IGMP (*Internet Group Multicast Protocol*)	RFC1112
IMAP (*Interactive Mail Protocol*)	RFC1176
Internet User's Glossary	RFC1392 (FYI18)
IP (*Internetworking Protocol*)	RFC791, RFC919, RFC922, RFC950, RFC1122
IPv6	RFC2292
MAIL	RFC822
MIME (*Multipurpose Internet Mail Extensions*)	RFC1494, RFC1495, RFC1496
NNTP (*Network News Transport Protocol*)	RFC977
POP (*Post Office Protocol*)	RFC1460
PPP (*Point to Point Protocol*)	RFC1570, RFC1331, RFC1332, RFC1134, RFC1172, RFC1334, RFC1548, RFC1663
RIP (*Routing Information Protocol*)	RFC1058, RFC1388
SLIP (*Serial Line Internetworking Protocol*)	RFC1055, RFC1144
SNMP (*Simple Network Management Protocol*)	RFC1098, RFC1157
STMP (*Simple Mail Transfer Protocol*)	RFC821, RFC1425, RFC1606, RFC1327
TCP (*Transmission Control Protocol*)	RFC793, RFC1122, RFC1323
TCP/IP	RFC1122, RFC1123
TELNET PROTOCOL	RFC854, RFC855, RFC1123
UDP (*User Datagram Protocol*)	RFC768, RFC1122

Tabla 4.3. Lista de los RFC más significativos

DOCUMENTACIÓN SOBRE INTERNET

Si se tiene interés en profundizar en el conocimiento de cómo funciona Internet, sus detalles están descritos en dos tipos de documentos: los llamados RFC (*Requests for Comments*, 'Petición de comentarios') y los llamados FYI (*For Your Information*, 'Para su información').

El protocolo TCP/IP es controlado por la ISOC (*Internet Society*, 'Sociedad Internet') a través de: IAB (*Internet Architecture Board*, 'Consejo de arquitectura de Internet') y el IETF (*Internet Engineering Task Force*, 'Grupo de trabajo de ingeniería de Internet'). Estos organismos redactan la documentación sobre TCP/IP en los informes conocidos como RFC.

Los documentos RFC empezaron a escribirse en 1969. El primero de ellos, RFC1, fue escrito por Steve Cracker y publicado por la Universidad de California UCLA. Los

RFC son gestionados por el editor de RFC. Hasta 1998 el editor fue Jonathan B. Postel, uno de los creadores de Internet y denominado 'hombre de oro' de la red. Desgraciadamente murió el 16 de octubre de 1998 cuando tan sólo contaba con 55 años. Su sucesor fue Joyce K. Reynolds.

Realmente, un RFC puede ser propuesto por cualquier persona. Simplemente debe proponer su nuevo estándar o su nuevo documento de trabajo al editor de RFC (*www.rfc-editor.org*). Si la propuesta es aceptada, se le asignará un número de RFC.

Los FYI son documentos que ofrecen una información muy útil acerca de Internet. Los FYI tienen una mayor audiencia que los RFC, ya que están dirigidos a los usuarios con menos experiencia. Realmente, los FYI son un subconjunto de las RFC.

Los STD (abreviatura de *standard*), son otro subconjunto de los RFC. Los STD hacen referencia a los RFC que de alguna manera fijan los estándares de Internet.

Los documentos RFC, FYI y STD están disponibles en muchos lugares de Internet, entre los que encuentran los siguientes:

www.rfc-editor.org
www.faqs.org/rfcs
www.rfc.net
www.rfc-ez.org

Transporte de voz en tiempo real

INTRODUCCIÓN

Anteriormente se han visto dos aspectos importantes de la comunicación de la voz sobre Internet: cómo se convierte la señal analógica de la voz en señal digital y qué mecanismos tiene Internet para hacer posible que una información digital se pueda transmitir a través de la red. El siguiente paso es ver cómo puede adaptarse Internet para transportar más eficientemente la voz.

Como ya se ha comentado, Internet se creó en un principio para transportar información que no tenía unas necesidades especiales en cuanto a tiempos de entrega. Lo importante era que absolutamente toda la información que se transmite en un extremo llegue al otro sin perder ni un solo bit. Para determinados tipos de información, como la voz o el vídeo, este extremo no es tan importante como el hecho de que la información sea entregada en el destino sin demora y como un flujo constante de datos. Esto es lo que se conoce como transporte de información en tiempo real.

Aunque el término tiempo real se suele usar también para hacer referencia a los sistemas que ofrecen tiempos de respuesta similares a los de la realidad (por ejemplo, un simulador de vuelo en tiempo real), lo fundamental de los protocolos de transporte de datos en tiempo real es que la información se lleve al destino con la misma cadencia de tiempo con la que se introduce en el origen y con un retardo mínimo. Estos protocolos se utilizan, fundamentalmente, en el transporte de las informaciones de audio y vídeo.

Como también se ha visto anteriormente, de los dos principales protocolos de la capa de transporte del modelo de Internet (TCP y UDP), el que permite el transporte de información en tiempo real es UDP. No obstante, este protocolo presenta algunos inconvenientes, como es el hecho de que no asegure que los paquetes se entreguen en el mismo orden en el que se recogieron. Por tanto, para hacer posible el transporte de información en tiempo real en Internet es necesario complementar el protocolo UDP con algún protocolo adicional. De esto se encargan los protocolos conocidos como RTP o protocolos de transporte en tiempo real.

EL PROTOCOLO DE TRANSPORTE EN TIEMPO REAL. RTP

En los años 90, el IETF creó un grupo de trabajo conocido como *Audio-Video Transport Working Group* (Grupo de trabajo para el transporte de audio y vídeo). Su finalidad era desarrollar un protocolo que permitiera el transporte de datos en tiempo real. En 1996 se publicaría el estándar RFC1886 que define el conjunto de protocolos RTP y RTCP. Este estándar quedaría actualizado en 2003 por el RFC3550 y complementado con la RFC3551 y RFC3711.

RTP es el acrónimo de *Real Time Transport Protocol* (Protocolo de transporte en tiempo real) y se encarga de añadirle a los paquetes UDP el número de secuencia, la marca de tiempo y la identificación del tipo de carga útil que transportan. Por su parte, RTCP (*RTP Control Protocol*, 'Protocolo de control RTP') se encarga de informar al remitente de la calidad de recepción y de la identidad de los interlocutores.

El protocolo RTP opera encima de UDP. Esto quiere decir que cuando UDP recibe los paquetes se los entrega al protocolo RTP, quien resuelve los posibles problemas que pudieran ocasionar la pérdida de paquetes o el cambio de orden de llegada. Para poder hacer esto, RTP le añade cierta información adicional a los paquetes, como son: un número de orden y el momento en el que el paquete salió del origen. La primera información se utiliza para detectar la pérdida o desorden de los paquetes, mientras que la segunda resulta útil para calcular parámetros de calidad como el retardo o las fluctuaciones de retardo (*jitter*). En realidad, RTP no hace nada para resolver estos problemas, pero los detecta e informa a los protocolos de capas superiores (a la aplicación de VoIP) para que puedan tomar las decisiones correspondientes.

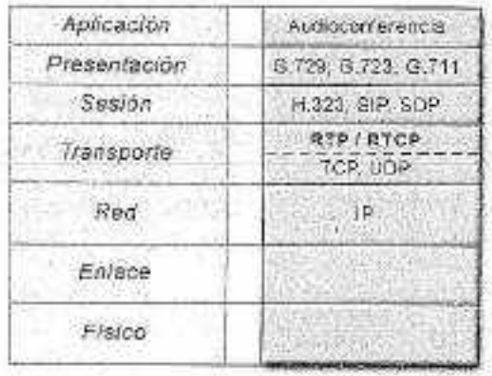

Figura 5.1. Protocolo RTP/RTCP dentro del modelo OSI

El protocolo RTCP se utiliza para el intercambio de información de control (número de paquetes perdidos, retardo, *jitter*, etc.) entre los distintos participantes de la sesión.

En circunstancias normales, el protocolo RTP y RTCP se utilizan conjuntamente, por lo que cuando se asigna un número de puerto a una sesión RTP, también se asigna otro para la sesión RTPCP correspondiente. Generalmente, el número de puerto de la sesión RTP es un número par entre 1.026 y 65.534 (por defecto el 5.004), mientras que el de la sesión RTCP es el número impar correlativo (por defecto el 5.005). No obstante, el uso del protocolo RTCP no es imprescindible para el funcionamiento de la comunicación, por lo que algunas aplicaciones de VoIP le ofrecen al usuario la posibilidad de desactivarlo, aunque esto suponga que no se intercambie información sobre la calidad de la transmisión.

Los protocolos RTP/RTCP pueden ser utilizados tanto para la transmisión de información multimedia *unicast* (unidifusión) como *multicast* (multidifusión). Esto quiere decir que pueden utilizarse tanto para transmisiones en las que hay un solo emisor y receptor (*unicast*), así como para aquellas en las que un mismo emisor transmite simultáneamente para distintos receptores (*multicast*).

CÓDIGO	CODIFICACIÓN	AUDIO/VIDEO	FRECUENCIA (Hz)	CANALES (Audio)	RFC
000	PCMU	Audio	8.000	1	RFC3551
001	Reservado				
002	G726-32	Audio	8.000	1	
003	GSM	Audio	8.000	1	RFC3551
004	G723	Audio	8.000	1	Kumar
005	DVI4	Audio	8.000	1	RFC3551
006	DVI4	Audio	16.000	1	RFC3551
007	LPC	Audio	8.000	1	RFC3551
008	PCMA	Audio	8.000	1	RFC3551
009	G722	Audio	8.000	1	RFC3551
010	L16	Audio	44.100	2	RFC3551
011	L16	Audio	44.100	1	RFC3551
012	QCELP	Audio	8.000	1	
013	CN	Audio	8.000	1	RFC3389
014	MPA	Audio	90.000	1	RFC3551, RFC2250
015	G728	Audio	8.000	1	RFC3551
016	DVI4	Audio	11.025	1	DiPol
017	DVI4	Audio	22.050	1	DiPol
018	G729	Audio	8.000	1	
019	Reservado				
020-023	Sin asignar	Audio			
024	Sin asignar	Video			
025	CelB	Video	90.000		RFC2029
026	JPEG	Video	90.000		RFC2435
027	Sin asignar	Video			
028	nv	Video	90.000		RFC3551
029-030	Sin asignar	Video			
031	H261	Video	90.000		RFC2032
032	MPV	Video	90.000		RFC2250
033	MP2T	Aud/Vid	90.000		RFC2250
034	H263	Video	90.000		Zhu
035-071	Sin asignar	Video			
072-076	Reservado				
077-095	Sin asignar				
096-127	Para uso de los tipos dinámicos				RFC3551

Tabla 5.1. Listado de números de identificación de codecs de RTP

Sistema codec utilizado

Una de las características de RTP es que incluye la identificación del sistema en el que está codificada la información de audio o video que contiene (el sistema codec utilizado en origen). El objetivo es que en el destino se sepa el codec que tienen que emplear para realizar la descodificación. En un principio, cuando se creó RTP sólo se disponía de un reducido número de codecs, por lo que a cada uno se le asignó un número de identificación de 8 bits (un número entre 0 y 255). No obstante, con el tiempo han ido apareciendo nuevos sistemas de codecs, por lo que 7 bits no parecían ser suficientes para identificarlos a todos ellos.

Para resolver esta situación se ideó el sistema dinámico de asignación de número de codecs. Esto quiere decir que al comienzo de la sesión RTP se acuerda el sistema codec a utilizar y el número de identificación que se le asigna. Por ejemplo, se acuerda que se va a utilizar el codec G729D con el número de tipo 122. Durante el tiempo que dura la sesión, siempre que se haga referencia al número 122 se identificará el sistema G729D. Cuando termina la sesión, el número 122 queda libre para ser utilizado de nuevo.

Los números de codecs del 0 al 95 identifican a sistemas de codificación concretos (conocidos como tipos estáticos), mientras que los números del 96 al 127 se utilizan para las asignaciones dinámicas. El listado de los tipos estáticos está recogido en la RFC1890. Por su parte, para que el sistema de asignación dinámica pueda identificar de forma inequívoca el codec a utilizar, es necesario que exista una lista de nombres de sistemas. Esta lista hace posible que todas las aplicaciones utilicen el mismo nombre para referirse a un codec concreto. Esta lista de nombres de codecs la gestiona el IANA.

Como las aplicaciones de VoIP no contienen todos los sistemas de codec, es posible que la aplicación del remitente disponga de codecs de los que no dispone la aplicación de destino. El proceso de negociación que incluye el sistema dinámico permite que ambas partes se pongan de acuerdo en el sistema codec a utilizar.

Por cierto, una curiosidad: entre los sistemas dinámicos de carga útil, existe uno conocido como RED (*Redundant*, 'Redundante'). Este sistema tiene la particularidad de enviar dos copias de cada muestra de sonido. En el resto de sistemas se envía una sola copia, por lo que, si se pierde un paquete, la aplicación de destino rellena lo mejor posible la falta de las muestras. En el caso de RED, cada paquete contiene sus muestras más una copia de las muestras del paquete precedente. Si se pierde un paquete, la aplicación sólo tiene que buscar las muestras perdidas en el paquete siguiente. Por cierto, las muestras redundantes enviadas pueden estar codificadas con un sistema distinto para ahorrar ancho de banda.

CÓDIGO	CODIFICACIÓN	AUDIO/ VIDEO	FRECUENCIA (Hz)	CANALES (Audio)	RFC
Dinámico	AMR	Audio	8.000	1	RFC3267
Dinámico	AMR-WB	Audio	16.000	1	RFC3267
Dinámico	DAT12	Audio		1	RFC3190
Dinámico	DSR-ES201108	Audio		1	RFC3557
Dinámico	EVRC	Audio	8.000	1	RFC3557
Dinámico	EVRC0	Audio	8.000	1	RFC3558
Dinámico	G7221	Audio	16.000	1	RFC3047
Dinámico	G726-16	Audio	8.000	1	RFC3551
Dinámico	G726-24	Audio	8.000	1	RFC3551
Dinámico	G726-32	Audio	8.000	1	RFC3551
Dinámico	G726-40	Audio	8.000	1	RFC3551
Dinámico	G729D	Audio	8.000	1	RFC3551
Dinámico	G729E	Audio	8.000	1	RFC3551
Dinámico	GSM-EFR	Audio	8.000	1	RFC3551
Dinámico	L8	Audio			RFC3551
Dinámico	RED	Audio			RFC2198, 3555
Dinámico	RTX	Audio			
Dinámico	VDVI	Audio		1	RFC3551
Dinámico	L20	Audio			RFC3190
Dinámico	L24	Audio			RFC3190
Dinámico	MPHA-LATM	Audio			RFC3016
Dinámico	MPA-ROBUST	Audio	90.000		RFC3119
Dinámico	Parityfec	Audio			RFC3009
Dinámico	SMV	Audio	8.000		RFC3558
Dinámico	SMV0	Audio	8.000		RFC3559
Dinámico	BMPEG	Video	90.000		RFC2343, 3555
Dinámico	BT656	Video	90.000		RFC2431, 3555
Dinámico	DV	Video	90.000		RFC3189
Dinámico	H263-1998	Video	90.000		RFC2429, 3555
Dinámico	H263-2000	Video	90.000		RFC2429, 3555
Dinámico	MP1S	Video	90.000		RFC2250, 3555
Dinámico	MP2P	Video	90.000		RFC2250, 3555
Dinámico	MP4V-ES	Video	90.000		RFC3016
Dinámico	RAW	Video	90.000		RFC3016

Tabla 5.2. Ejemplos de nombres de codec del sistema dinámico

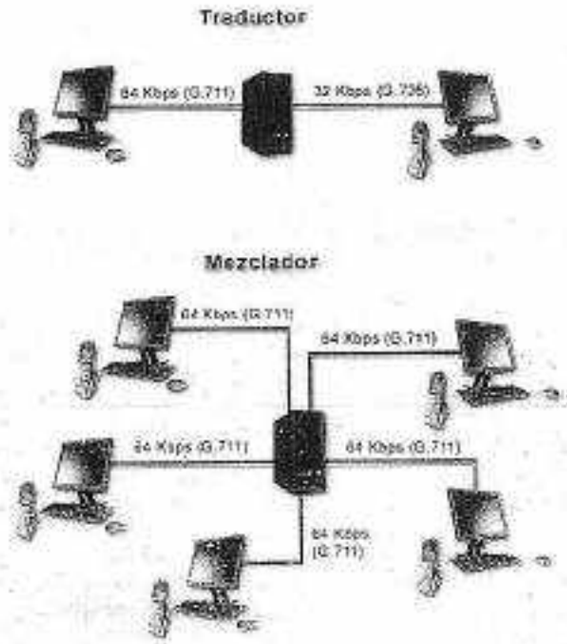

Figura 5.2. Ejemplo de traductor y mezclador

Mezcladores y traductores

Lo normal es que una comunicación de voz se lleve a cabo entre un remitente y un destinatario que utilizan el mismo sistema codec, no obstante, se puede dar la circunstancia de que el remitente y el destinatario no dispongan de codecs. También puede ocurrir que participe más de un destinatario con codecs distintos. En estos casos, el protocolo RTP considera la posibilidad de que existan aplicaciones intermedias que lleven a cabo las conversiones necesarias. Las opciones actuales son dos:

- **Traductor**: se trata de una aplicación que realiza una conversión de sistema de codificación. Si los participantes de la comunicación no utilizan el mismo sistema o el mismo ancho de banda, el traductor se encarga de realizar la adaptación.
- **Mezclador**: se trata de una aplicación que recoge distintos flujos de sonido de distintas fuentes y los convierte en un flujo único. Un ejemplo de esto sería una audioconferencia, donde cada participante recibe el sonido procedente del resto de participantes.

Una misma aplicación podría hacer el trabajo del mezclador y traductor. Por ejemplo, si los participantes de una audioconferencia utilizan sistemas de codec distintos, el mezclador/traductor podría realizar las conversiones necesarias para entregar el sonido a cada participante en su propio sistema.

Identificación de los participantes

Para identificar a los distintos participantes que forman parte de una comunicación se utiliza un número de 32 bits. Este número no está relacionado con la dirección IP del remitente, sino que lo genera cada remitente de forma aleatoria. Se supone que, dado que con 32 bits se pueden dar más de 4.000 millones de combinaciones y que el número de participantes es reducido, la probabilidad de que dos participantes generen el mismo número de identificación es extremadamente baja. No obstante, se cuenta con mecanismos de prevención de coincidencias (colisión) para prevenir estos casos.

El número que identifica a cada fuente se conoce como SSRC (*Synchronization Source*, 'Origen de sincronización'). Este dato aparece en la cabecera de los paquetes RTP, no obstante, cuando en una audioconferencia intervienen múltiples participantes con un mezclador intermedio, el campo SSRC identifica al mezclador, identificándose cada participante en una lista de campos que se incluye a continuación y que se conoce con el nombre de lista de contribuyentes o CSRC (*Contributing Source*, 'Origen de la contribución').

V (2b)
P
X
CC (4 bits)
M
PT: Tipo carga útil (7b)
Número de secuencia (16 bits)
Contador de tiempo o timestamp (32 bits)
SSRC: Identificación del origen (32 bits)
CSRC: Lista de contribuyentes (de 0 a 16 entradas de 32 bits)
Información específica (16 bits)
Longitud (1
Extensión de la cabecera
Extensión de la cabecera (opcional)
Carga útil (audio, video)
Relleno
Tamaño del relleno (8 bits)
V=Versión, P=Relleno, X=Extensión, CC=Cuenta CSRC, M=Marcador

Figura 5.3. Formato de los paquetes RTP

Formato de los paquetes RTP

Los paquetes RTP se componen de una cabecera de 128 bits y de un cuerpo que contiene la información de la voz codificada (o vídeo). El contenido del cuerpo se conoce como carga útil o *payload*. El tamaño del cuerpo puede ser distinto dependiendo del tipo de carga útil utilizada (tipo de codec), no obstante, siempre debe ser múltiplo de 32 bits.

Determinados tipos de carga útil necesitan incluir más información de control de la que cabe en la cabecera. Para estos casos existen dos alternativas: esta información puede venir incluida como parte de la carga útil (en los primeros n octetos) o se puede utilizar una extensión de la cabecera que se colocará a continuación de la cabecera normal y antes de la carga útil. El formato de la extensión de la cabecera incluye un campo de 16 bits para indicar la longitud de la misma.

Las informaciones que se incluyen en la cabecera de los paquetes RTP son las siguientes:

- **Versión** (V). Se trata de dos bits que indican la versión del protocolo RTP utilizado. La versión más reciente actualmente es la 2.
- **Relleno** (*Padding*, P). Se trata de un bit que indica que la carga útil incluye octetos de relleno al final. El último octeto de relleno indica el número total de octetos de relleno existentes. Como el tamaño de la carga útil debe ser múltiplo de 32 bits, el relleno, de haberlo, tendrá un máximo de 24 bits. El bit de relleno se pone a 1 para indicar que existe un relleno.
- **Extensión** (X). Se trata de un bit que indica que la cabecera utiliza el formato extendido.
- **Cuenta CSRC** (CC). Utiliza 4 bits para indicar el número de identificadores CSRC que se añaden al final de la cabecera fija. Este campo viene a identificar el número de participantes en la comunicación.
- **Marcador** (*Marker*, M). Se trata de un campo de un bit que la RFC1889 deja para que la carga útil lo utilice libremente y que la RFC1890 utiliza para que las aplicaciones que no envían información en los periodos de silencio fijen este bit a uno en el primer paquete después de un periodo de silencio.
- **Tipo de carga útil** (*Payload Type*, PT). Se trata de un campo de 7 bits que contiene el número que identifica el codec utilizado en la carga útil.
- **Número de secuencia** (*Sequence Number*). Se trata de un campo de 16 bits que utiliza el remitente para identificar el orden secuencial de envío de los paquetes. Este número le permite al destinatario detectar la posible pérdida o desorden de los paquetes. El número de secuencia del primer paquete se genera de forma aleatoria, incrementándose en una unidad para cada uno de los paquetes siguientes.
- **Contador de tiempo** (*Timestamp*). Se trata de un campo de 32 bits que indica el instante en el que se generó la primera muestra de la carga útil. El campo

muestra un número entero donde cada unidad es equivalente al periodo de la frecuencia de muestreo utilizada o, lo que es lo mismo, al periodo de tiempo de cada muestra. Por ejemplo, si la frecuencia de muestreo es 8.000 Hz, cada unidad representará el equivalente a 0,125 milisegundos (1/8.000). Por otro lado, si cada paquete contiene 5 muestras de voz y el contador de tiempo incluido en el paquete es 31, el siguiente paquete debería mostrar un contador de tiempo igual a 36. Para una frecuencia de 8.000 Hz, esto sería equivalente a indicar que entre la primera muestra del primer paquete y la primera muestra del segundo paquete ha transcurrido un tiempo de 0,625 milisegundos. El contador de tiempo se utiliza para identificar la fluctuación de retardo (*jitter*).

- **Identificación del origen** (*Synchronization Source*, SSRC). Se trata de un campo de 32 bits que identifica al remitente o a la aplicación intermedia utilizada (mezclador).
- **Lista de contribuyentes** (*Contributing Source*, CSRC). Cuando se utiliza un mezclador, el campo SSRC identifica al mezclador, utilizándose una lista de campos de 32 bits para identificar a cada una de las fuentes de sonido (o vídeo). El número de fuentes de la lista se especifica en el campo CC. Como el campo CC tiene 4 bits, el máximo número de fuentes que permite RTP es 15.

INTERCAMBIO DE INFORMACIÓN. RTCP

El protocolo RTCP facilita el intercambio periódico de información entre los participantes de la sesión. La finalidad de RTCP es informar a la fuente del sonido (o vídeo) de la calidad con la que está llegando al destino. Esta información puede ser utilizada por la aplicación o por el operador del servicio para detectar y corregir posibles problemas.

Una información incluida en RTCP que no se incluye en los paquetes RTP es el nombre CNAME (*Canonical Name*, 'Nombre canónico') de los participantes de la sesión. RTP utiliza el número SSRC para identificar a los participantes, pero este número puede cambiar de una sesión a otra, o incluso el mismo participante podría generar distintos SSRC en una misma sesión (por ejemplo, cuando emite audio y vídeo simultáneamente). El nombre CNAME es único para cada participante y tiene la forma *usuario@host*. Este nombre no está relacionado con ninguna dirección de correo electrónico, sino que lo crea automáticamente la aplicación componiendo el nombre del usuario y la identificación del ordenador donde se encuentra. Por ejemplo, *juan@192.0.2.89* sería un CNAME. A este nombre también se lo conoce como nombre oficial o nombre único.

TIPO	NOMBRE	DESCRIPCIÓN	RFC
192	FIR	Full intra-frame request	RFC2032
193	NACK	Negative acknowledgement	RFC2032
200	SR	Sender report	RFC3551
201	RR	Receiver report	RFC3551
202	SDES	Source description	RFC3551
203	BYE	Goodbye	RFC3551
204	APP	Application-defined	RFC3551
205	RTPFB	Generic RTP feedback	
206	PSFB	Payload-specific	
207	XR	Extended report	RFC3611

Tabla 5.4. Tipos de paquetes RTCP

Se tienen distintos tipos de paquetes RTCP:

- **Informe de emisor** (SR, *Sender Report*). Se utiliza para informar de las estadísticas de los participantes que son emisores activos.
- **Informe de receptor** (RR, *Receiver Report*). Se utiliza para informar de las estadísticas de los participantes que son sólo receptores del flujo, no emisores (sólo escuchan).
- **Descripción de la fuente** (SDES, *Source Description*). Contiene la descripción de la fuente, incluyendo el nombre CNAME, así como información de carácter personal como el nombre, correo electrónico o número de teléfono del participante.
- **Fin** (*Bye*). Indica el final de la participación en la sesión.
- **Funciones específicas de la aplicación** (APP, *Application-Specific Functions*). Los paquetes APP permiten enviar información específica de una determinada aplicación o tipo de flujo.

Las especificaciones de RTCP indican que estos paquetes deben enviarse agrupados, de forma que cada grupo incluya, al menos, un paquete de informe (SR o RR) y un paquete SDED. Por tanto, para enviar un paquete de fin de la participación, además habría que incluir, por ejemplo, un paquete SR y otro SDED. Además, el envío de paquetes RTCP debe cumplir otras condiciones como: las estadísticas de recepción (SR o RR) se deben enviar tan frecuentemente como lo permita el ancho de banda, los nuevos receptores deben recibir el CNAME de la fuente tan pronto como sea posible, el intervalo entre paquetes RTCP debe ser mayor de 5 segundos (este intervalo debe ser calculado aleatoriamente por cada participante para evitar que todos lo hagan al mismo tiempo), etc.

Informe de emisor. SR

El informe de emisor o SR lo utilizan los participantes de la sesión que son emisores activos de paquetes RTP. Los paquetes SR suelen aprovecharse para adjuntar mensajes RR con información sobre los paquetes RTP que se reciben de otros participantes.

Los paquetes SR tienen cuatro secciones distintas:

- Información de cabecera (32 bits)
- El informe de emisor propiamente dicho (160 bits)
- Un informe de receptor RR (192 bits) por cada fuente emisora que se ha escuchado desde el último informe. Si no se ha recibido nada no se adjuntará ningún RR.
- Determinados perfiles específicos necesitan, además, de una extensión de cabecera que sería añadida al final del paquete.

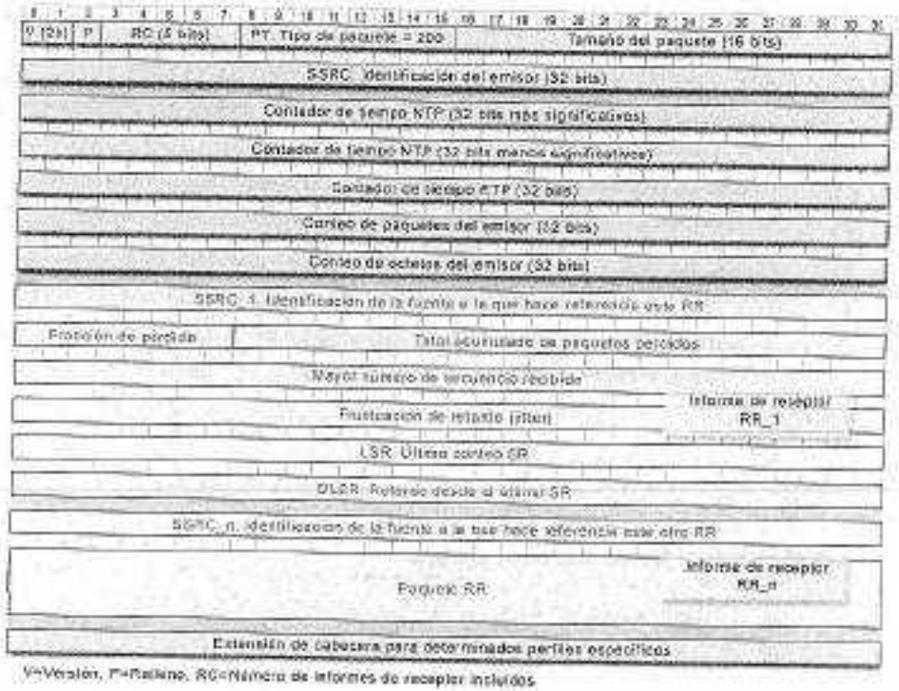
V (2b)
P
RC (5 bits)
PT. Tipo de paquete = 200
Tamaño del paquete (16 bits)
SSRC. Identificación del emisor (32 bits)
Contador de tiempo NTP (32 bits más significativos)
Contador de tiempo NTP (32 bits menos significativos)
Contador de tiempo RTP (32 bits)
Conteo de paquetes del emisor (32 bits)
Conteo de octetos del emisor (32 bits)
SSRC_1. Identificación de la fuente a la que hace referencia este RR
Fracción de pérdida
Total acumulado de paquetes perdidos
Mayor número de secuencia recibido
Fluctuación de retardo (jitter)
Informe de receptor RR_1
LSR. Último conteo SR
DLSR. Retardo desde el último SR
SSRC_n. Identificación de la fuente a la que hace referencia este otro RR
Paquete RR
Informe de receptor RR_n
Extensión de cabecera para determinados perfiles específicos
V=Versión, P=Relleno, RC=Número de informes de receptor incluidos

Figura 5.4. Formato de los paquetes SR

La cabecera de los paquetes SR dispone de los siguientes campos:

- Versión (V). Se trata de dos bits que indican la versión del protocolo RTP utilizada. La versión actual es la 2.
- Relleno (P, *padding*). Se trata de un bit para indicar que al final del paquete hay octetos de relleno. El último de estos octetos indica el total de ellos.
- Número de informes de receptor incluidos en el paquete (RC, *Reception Report Count*). Este campo tiene 5 bits, por lo que el número máximo de RR que se pueden incluir es 31. El mínimo sería cero.
- Tipo de paquete (PT, *Packet Type*). Son 8 bits que indican el tipo de paquete que se está enviando. En el caso de los paquetes SR tomaría el valor 200.
- Longitud del paquete (*length*). Indica el número de grupos de 32 bits que forman el paquete RTCP, sin incluir la cabecera. O lo que es lo mismo, número total de grupos menos uno.
- SSRC del remitente de este paquete RTCP.

Los campos correspondientes del informe de emisor son los siguientes:

- **Contadores de tiempo NTP** (*Network Time Protocol*, RFC1305). Se trata de un campo de 64 bits que indica el momento en el que se emitió este informe SR. Lo que aquí se indica es el tiempo transcurrido en segundos desde el 1 de enero de 1900 (GMT). Los 32 bits más significativos representan el número de segundos, mientras que los 32 bits menos significativos representan las fracciones de segundos, consiguiéndose una precisión de 200 picosegundos.
- **Contador de tiempo RTP** (*RTP timestamp*). Es un contador de tiempo de 32 bits que emplea el mismo formato que los contadores RTP. La inclusión de dos contadores de tiempo en el mismo informe le permite al destinatario sincronizarse mejor con el emisor.
- **Conteo de paquetes del emisor**. Indica el número total de paquetes RTP transmitidos por el emisor desde el comienzo de la sesión. Por tanto, este conteo de paquetes es acumulativo.
- **Conteo de octetos del emisor**. Indica el número total de octetos de carga útil (sin incluir cabecera ni relleno) incluidos en los paquetes RTP que ha enviado este emisor desde el comienzo de la sesión.

A la información del emisor le pueden seguir uno o más bloques RR. El incluir estos bloques en el mismo paquete SR ahorra ancho de banda al no tener que enviar paquetes RR independientes con su propia cabecera. No obstante, si no existen bloques RR que transmitir, simplemente se pondrá a cero el campo RC de la cabecera y no se añadirá ningún bloque RR.

Informe de receptor. RR

Todos los participantes que reciben información envían como respuesta los correspondientes informes de receptor RR para mantener informado al emisor de la calidad de recepción. Como hemos visto, en el caso de que el receptor sea también un emisor activo, los bloques RR se podrán incluir al final del paquete SR.

Al igual que los paquetes SR, los RR disponen de su cabecera, con el mismo formato que la de los paquetes SR, y de la posibilidad de añadir una extensión de cabecera. En este caso, el campo PT (tipo de paquete) toma el valor 201.

V (2b)
P
RC (5 bits)
PT. Tipo de paquete = 201
Tamaño del paquete (16 bits)
SSRC. Identificación del emisor (32 bits)
SSRC_1. Identificación de la fuente a la que hace referencia este RR (32 bits)
Fracción de pérdida (8 bits)
Total acumulado de paquetes perdidos (24 bits)
Mayor número de secuencia recibido (32 bits)
Fluctuación de retardo o jitter (32 bits)
LSR. Último conteo SR (32 bits)
DLSR. Retardo desde el último SR (32 bits)
SSRC_n. Identificación de la fuente a la que hace referencia este otro RR (32 bits)
Paquete RR
Extensión de cabecera para determinados perfiles específicos
V=Versión, P=Relleno, RC=Número de informes de receptor incluidos

Figura 5.5. Formato de los paquetes RR

A continuación de la cabecera se incluyen tantos bloques RR como sean necesarios (con un máximo de 31). El formato de estos bloques es el siguiente:

- SSRC_n. Es el identificador SSRC del emisor de la información cuyos datos de evaluación se envían a continuación.
- Fracción de pérdida. Es un campo de 8 bits que indica el porcentaje de paquetes RTP que se han perdido desde el último informe. El valor indica el numerador de una fracción donde el denominador es 256. Por ejemplo, si el valor incluido es 64, esto indicará que se han perdido 64/256 = 25%.
- Total acumulado de paquetes perdidos. Indica el número total de paquetes RTP perdidos desde el comienzo de la sesión.

- Mayor número de secuencia recibido. Se trata del número de secuencia del último paquete RTP que se ha recibido desde este emisor. Los 16 bits de menor peso contienen el último número de secuencia recibido, mientras que los 16 bits de mayor peso indican simplemente un conteo cíclico.
- Fluctuación de retardo (*interarrival jitter*). Se trata de una estimación de la variación del retardo entre dos paquetes RTP. La información se ofrece con las mismas unidades que el contador de tiempo RTP.
- Último conteo SR (LSR, *Last SR Timestamp*). Si SSRC_n es un emisor activo que ha enviado anteriormente un informe de emisor, en este campo se indican los 32 bits centrales de los 64 que se compone el contador de tiempo NTP incluido en el último informe SR recibido de SSRC_n. La utilidad de este campo es informar que se ha recibido bien su último informe SR.
- Retardo desde el último SR (DLSR, *Delay Since Last SR*). Si SSRC_n es un emisor activo que ha enviado anteriormente un informe de emisor, en este campo se indica el periodo de tiempo entre la recepción del último informe SR y el envío de este informe RR. Este tiempo se indica en unidades de 1/65.536 segundos.

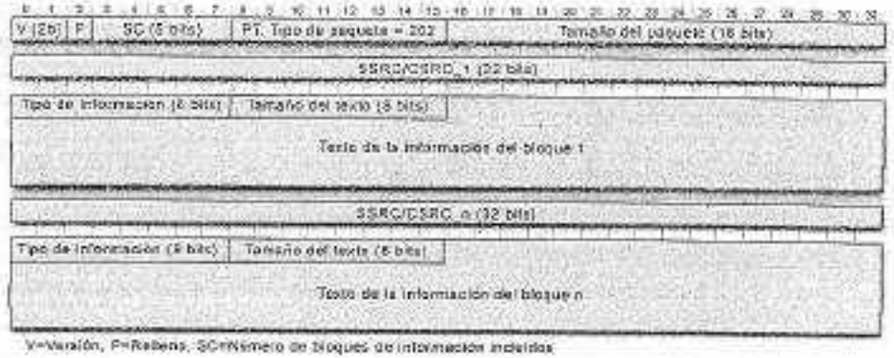
V (2b)
P
SC (5 bits)
PT. Tipo de paquete = 202
Tamaño del paquete (16 bits)
SSRC/CSRC_1 (32 bits)
Tipo de información (8 bits)
Tamaño del texto (8 bits)
Texto de la información del bloque 1
SSRC/CSRC_n (32 bits)
Tipo de información (8 bits)
Tamaño del texto (8 bits)
Texto de la información del bloque n
V=Versión, P=Relleno, SC=Número de bloques de información incluidos

Figura 5.6. Formato de los paquetes SDES

Descripción de la fuente. SDES

Los paquetes SDES facilitan la identificación de los participantes de la sesión mediante el envío del nombre CNAME. No obstante, determinadas aplicaciones requieren intercambiar también otros datos como nombre, correo electrónico o número de teléfono del participante. Los paquetes SDES permiten intercambiar también este tipo de datos.

El formato de los paquetes SDES se compone de una cabecera seguida de tantos bloques de información como sea necesario. La cabecera tiene el mismo formato que la de los campos SR o RR, pero en este caso, el tipo de paquete es el 202 y, en vez de enviarse el campo RC con el número de informes de receptor incluidos, se envía el campo SC (*Source Count*) indicando el número de bloques de información que se incluyen en este paquete.

TIPO	NOMBRE	DESCRIPCIÓN	RFC
0	END	Fin de la lista SDES	RFC3551
1	CNAME	Nombre canónico	RFC3551
2	NAME	Nombre del usuario	RFC3551
3	EMAIL	Correo electrónico del usuario	RFC3551
4	PHONE	Número de teléfono del usuario	RFC3551
5	LOC	Localización geográfica	RFC3551
6	TOOL	Nombre de la aplicación	RFC3551
7	NOTE	Aviso sobre el emisor	RFC3551
8	PRIV	Extensiones privadas	RFC3551
9	H323CADDR	Dirección H323 llamable	Kumar

Tabla 5.3. Tipos de informaciones incluidas en SDES

Cada bloque de información contiene un valor SSRC o CSRC, seguido por una o más piezas de información. Cada pieza de información se compone de tres campos: tipo de información (8 bits), longitud en octetos del texto de la información (8 bits) y el texto de la información (con el tamaño indicado en el campo anterior, con un máximo de 255 octetos). Por cierto, la estructura de las informaciones SDES se describe en RFC 1889.

Figura 5.7. Formato de los paquetes Bye

Fin de la participación. Bye

El paquete Bye se utiliza para indicar que uno o más participantes dejan de estar activos. Este paquete puede incluir también una cadena de texto con la razón por la que se abandona la sesión.

El formato del paquete Bye se compone de una cabecera, seguida por uno o varios SSRC/CRSC para terminar con un campo de 8 bits que indica la longitud del texto que se incluye a continuación (en octetos) describiendo la razón por la que se abandona la sesión. En la cabecera, el tipo de paquete es el 203 y el campo SC (*Source Count*) indica el número de SSRC/CRSC incluidos en el paquete.

Funciones específicas de la aplicación. APP

El tipo de paquete APP se utiliza para permitir el desarrollo de nuevas funcionalidades o tipos de paquetes RTP. Los paquetes APP los utilizan libremente las aplicaciones cuando necesitan transmitir alguna información no considerada en ninguno de los tipos de paquetes estándares RTP.

El formato del paquete APP incluye una cabecera, seguida por SSRC/CRSC, el campo nombre (32 bits) y los datos específicos a transmitir. En la cabecera, el tipo de paquete es el 204 y el campo SC o RC se sustituye por el campo subtipo. Tanto el campo nombre como el campo subtipo los definen las aplicaciones para cubrir sus propias necesidades.

Figura 5.8. Formato de los paquetes APP

CALCULAR LOS PARÁMETROS DE CALIDAD

Los paquetes RTCP contienen la información necesaria para realizar las estimaciones pertinentes de los parámetros de calidad de cada una de las comunicaciones. Los cálculos a realizar y el uso que se le dé a estos resultados dependen de la aplicación concreta de comunicación y del proveedor del servicio. Lo que no cabe duda es que estos resultados pueden utilizarse para tomar decisiones que permitan mejorar la calidad de la comunicación. Por ejemplo, se podría modificar el sistema codec utilizado o su ancho de banda.

Por ejemplo, supongamos que un emisor activo A envía un paquete SR que llega a un receptor B y que éste responde con un paquete RR. El primer paquete incluirá un campo NTP que indica el instante T1 en el que sale de A. Por su lado, el paquete RR incluye el campo DLSR que indica la diferencia de tiempo desde que B recibió el paquete SR (T2) y envió su paquete RR (T3), así como el campo LSR que le sirve a A para relacionar este paquete concreto RR con el paquete concreto SR que él envió.

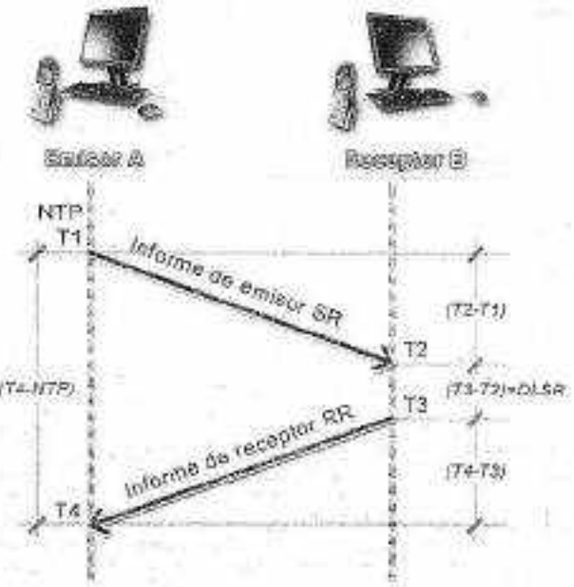

Figura 5.9. Cálculo del retardo de ida y vuelta

Cuando el paquete RR llega a A (T_4) se puede conocer el retardo de ida y vuelta comprobando los campos anteriores y haciendo el siguiente cálculo:

$$(T_4-T_1)+(T_3-T_2) = T_4-(T_3-T_2)-T_1 = T_4-\mathbf{DLSR}-\mathbf{NTP}$$

Para calcular la fluctuación de retardo (*jitter*) se tiene el instante en el que cada paquete RTP sale del remitente y el instante en el que llega al destinatario. Basta con comprobar las desviaciones.

Establecimiento de la comunicación con H.323

INTRODUCCIÓN

En los capítulos precedentes se ha explicado cómo se puede digitalizar la voz y cómo se transporta esta información en tiempo real a través de la red IP. Lo que viene a continuación es resolver cómo pueden dos usuarios cualesquiera de una red IP (corporativa o Internet) establecer una comunicación de voz entre ellos. ¿Cómo se identifican los teléfonos dentro de la red IP? ¿cómo se sabe si el terminal está disponible? ¿cómo se da por terminada la llamada? ¿cómo se decide la codificación a emplear? o ¿cómo se pueden facturar dichas llamadas?. Dicho de otra forma: cómo se ponen de acuerdo los terminales participantes para que la comunicación se pueda establecer, desarrollar y dar por terminada con normalidad. La respuesta es: gracias a los sistemas de señalización.

Todos estamos acostumbrados al funcionamiento del servicio telefónico tradicional: descolgamos el teléfono, marcamos el número con el que deseamos comunicar, escuchamos el tono de llamada, esperamos a que contesten y hablamos. Cuando queremos dar por terminada la comunicación, simplemente, colgamos el auricular o pulsamos el botón rojo del terminal.

Desde el punto de vista de la red telefónica, existen dos tipos de señalización:

- Señalización entre el terminal y la central telefónica. Esta señalización se utiliza, fundamentalmente, para transmitir el número al que se desea llamar o el número que llama y recibir los tonos de ocupado, llamada, etc. Hoy en día se suele utilizar la señalización multifrecuencia (conocida como DTMF o *Dual-Tone Multifrequency*, 'Doble tono multifrecuencia') para las líneas analógicas y la señalización de canal D (Q.931) para las líneas RDSI.
- Señalización entre las centrales que forman la red. En general, esta señalización se utiliza para que progrese el establecimiento de la llamada o se liberen los recursos al terminar. El sistema más común es el SS7 (sistema de señalización número 7 de la UIT-T).

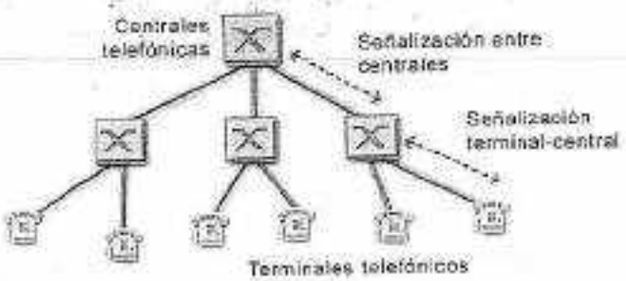

Figura 6.1. Señalización en la red telefónica

La finalidad principal de un sistema de señalización es la siguiente:

- Identificar los terminales que forman parte de la red (grupos de usuarios).
- Definir los procedimientos necesarios para que un usuario pueda iniciar una llamada, responder a la misma o darla por terminada, así como identificar cada uno de los estados posibles (ocupado, ausente, no existe, etc.).
- Establecer los parámetros de la comunicación compatibles con los terminales que participan.
- Permitir la utilización de servicios de valor añadido (desvío de llamada, conferencia a tres, llamada en espera, etc.).
- Permitir la gestión de usuario desde el punto de vista del operador (registro de usuario, duración de la llamada, facturación, etc.).

En el caso de la telefonía IP nos encontramos con la particularidad de que la red IP es una red abierta que no es propiedad de un operador ni se utiliza en exclusividad para este servicio. Esto obliga a que el sistema de señalización coexista con otras aplicaciones y deba ser compatible con la filosofía de red abierta. Por otro lado, un servicio de telefonía IP puede desplegarse con una cobertura limitada, dentro de una red corporativa (red de área local o Intranet), o con una cobertura amplia, dentro de Internet.

SISTEMAS DE SEÑALIZACIÓN DE VOIP

La historia de la telefonía IP ha traído distintos sistemas de señalización. El primero de ellos fue completamente propietario. Quiere esto decir que sólo podían comunicarse los terminales del mismo fabricante. Esta falta de interoperatividad impedía el despliegue de la VoIP, lo que llevó a la UIT-T a desarrollar un sistema estándar de señalización para la telefonía IP: la recomendación H.323 (1996). Los sistemas de señalización de la telefonía tradicional son de la UIT-T, así que parece lógico que también desarrollaran este nuevo sistema IP.

Otro organismo que tiene mucho que decir en los sistemas de señalización de telefonía IP es la IETF. Ellos desarrollaron el protocolo IP, por lo que también son buenos candidatos a proponer un sistema de señalización que funciona sobre IP. Su propuesta se llama SIP (*Session Initiation Protocol*, 'Protocolo de inicio de sesión'), publicándose su primera versión en 1999 (RFC2543).

Además de los sistemas de señalización anteriores, existen otros sistemas desarrollados por compañías privadas con la idea de implementar características que no recogen los sistemas normalizados. Algunos de estos sistemas son: SCCP (*Skinny Client Control Protocol*, 'Protocolo ligero de control de clientes') de Cisco, IAX de Digitum o Unistim de Nortel. El resultado es que existe una gran variedad de sistemas de

señalización que tienen en común el 75% de sus características. En cualquier caso, de todos ellos, los más generalizados siguen siendo los normalizados por la UIT-T y el IETF: H.323 y SIP.

PROTOCOLO	RESPONSABLE	ESCENARIO
H.323	*UIT-T*	*Telefonía y vídeo*
SIP	*IETF*	*Telefonía, mensajería instantánea y vídeo*
IAX	*Digium*	*Telefonía*
SCCP	*Cisco System*	*Telefonía (switch to endpoint)*
MEGACO/ H.248	*IETF y UIT-T*	*Telefonía, control del gateway*
MGCP	*IETF*	*Telefonía, control del gateway*

Tabla 6.1. Protocolos de señalización de VoIP

El nacimiento de H.323

La primera versión de H.323 se publicó en 1996 con el objetivo de permitir las comunicaciones multimedia (voz y vídeo) sobre redes IP de área local. El nombre original de H.323 era 'Sistemas y equipos de telefonía visual para redes de área local'. Debido a que esta versión no contaba con la funcionalidad necesaria para trabajar en entornos más amplios, la UIT-T sacó en 1998 la versión 2 con un nuevo título: 'Sistemas de comunicación multimedia basados en paquetes'. A pesar de su buena aceptación inicial, han sido necesarias nuevas versiones para adaptarse a las continuas exigencias del mercado: versión 3 (septiembre de 1999), versión 4 (noviembre de 2000), versión 5 (julio de 2003) y versión 6 (junio de 2006).

El estándar H.323 es completamente compatible con los sistemas de telefonía tradicional, permitiendo, incluso, la señalización extremo a extremo entre terminales conectados a redes distintas (recomendación H.320 y H.324). Otra de las principales ventajas de H.323 es que es bastante ligero y rápido al ocupar la red el tiempo imprescindible para establecer la comunicación.

El software de multiconferencia de Microsoft conocido como Netmeeting, así como MultiVantag de Avaya o los sistemas Meridian de centralitas IP de Nortel utilizan H.323. Algunos otros fabricantes han adaptado este estándar a sus propias necesidades específicas, desarrollando sistemas propietarios muy similares a H.323 pero no completamente compatibles.

LA ARQUITECTURA DE H.323

El sistema de señalización H.323 basa su funcionamiento en la existencia de cuatro elementos:

- **Terminal**. Se trata del terminal telefónico que utiliza directamente el usuario para establecer la comunicación de voz e vídeo. Viene a ser lo que en la red telefónica tradicional es el teléfono. En el caso de H.323, este teléfono puede ser un software instalado en un PC o un equipo hardware específico.
- ***Gateway*** (pasarela). Este equipo hace posible que una red H.323 (por ejemplo, una red corporativa de VoIP) se interconecte con otro tipo de red de voz (por ejemplo, la red telefónica tradicional). Un lado del *gateway* se conecta a la red H.323 y el otro a la otra red. El *gateway* se encarga de interpretar cada una de las señalizaciones y de realizar las traducciones necesarias para que las comunicaciones iniciadas en una de las redes se pueda completar en la otra y viceversa.
- **Unidad de Control Multipunto** o MCU (*Multipoint Control Unit*). Se trata de un equipo que gestiona conferencias en las que participan más de dos terminales. Estas multiconferencias se pueden organizar de forma centralizada o descentralizada. En el primer caso, todos los terminales se comunican con el MCU, quien se encarga, no sólo de controlar la comunicación, sino de distribuir la información de voz/vídeo. En el segundo caso, el MCU sólo realiza el control de la comunicación, utilizándose la tecnología *multicast* para el intercambio de la información multimedia (voz y vídeo).

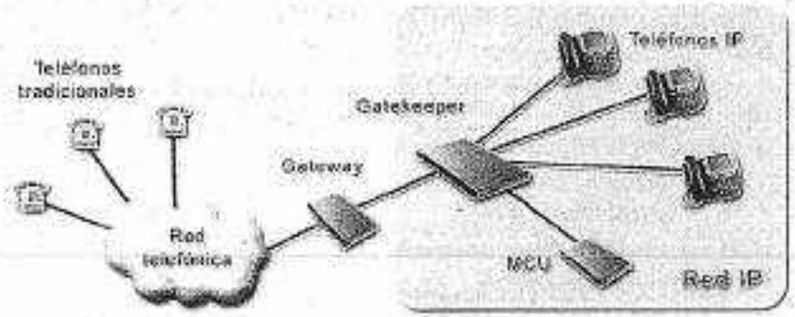

Figura 6.2. Modelo H.323

- ***Gatekeeper*** (aunque literalmente significa portero o recepcionista, sería más apropiado hablar de mediador o centralita IP). Es un equipo que realiza una labor equivalente a la de una centralita telefónica tradicional. De hecho, a veces se hace referencia al *gatekeeper* como centralita software, *softswitch* o *softPBX*.

En el caso de H.323, el *gateway* está especialmente pensado para interconectar la red H.323 (soportada sobre una red IP corporativa o Internet) con la red telefónica tradicional. Esto quiere decir que el *gateway* entenderá, tanto la señalización y codificación propia de H.323 como las de la red telefónica en sus distintas modalidades:

- Líneas telefónicas analógicas conectadas a la red telefónica básica
- Líneas telefónicas digitales de la red telefónica básica (circuitos digitales de 64 kbps que cumplen H.324 y G.711).
- Líneas telefónicas digitales de la red digital de servicios integrados (RDSI o ISDN en su acrónimo en inglés) que cumple H.320.

Aunque, desde el punto de vista de la red H.323, el *gateway* es un terminal más, en realidad se trata de un equipo que proporciona muchos servicios distintos, entre los más importantes se encuentran los siguientes:

- Traducción entre formatos de transmisión (por ejemplo, de H.225 a H.221)
- Traducción entre procedimientos de comunicación (por ejemplo, de H.245 a H.242)
- Traducción entre los codecs de audio y vídeo utilizado por ambas redes.
- Procesa la configuración de la llamada en ambos lados de la comunicación.

En cualquier caso, no todos los equipos *gateway* son iguales, cada fabricante puede diseñar su equipo para, por ejemplo, atender un número distinto de llamadas simultáneas o incluir determinados codecs de audio, vídeo o datos.

Unidad de control multipunto

La unidad de control multipunto o MCU es un equipo especialmente pensado para el establecimiento de comunicaciones donde intervienen más de dos participantes. Es lo que se conoce generalmente como multiconferencias.

El MCU se compone de dos partes:

- **MP** (*Multipoint Processor*, 'Procesador multipunto'). Se encarga de mezclar, conmutar o combinar las informaciones multimedia (voz, vídeo o datos) que vienen de cada terminal, así como de convertir los codecs de forma que cada terminal reciba la información multimedia en el formato que puede entender.
- **MC** (*Multipoint Controller*, 'Controlador multipunto'). Se encarga de la señalización y control de los equipos finales involucrados. Para ello utiliza las recomendaciones H.225.0 y H.245.

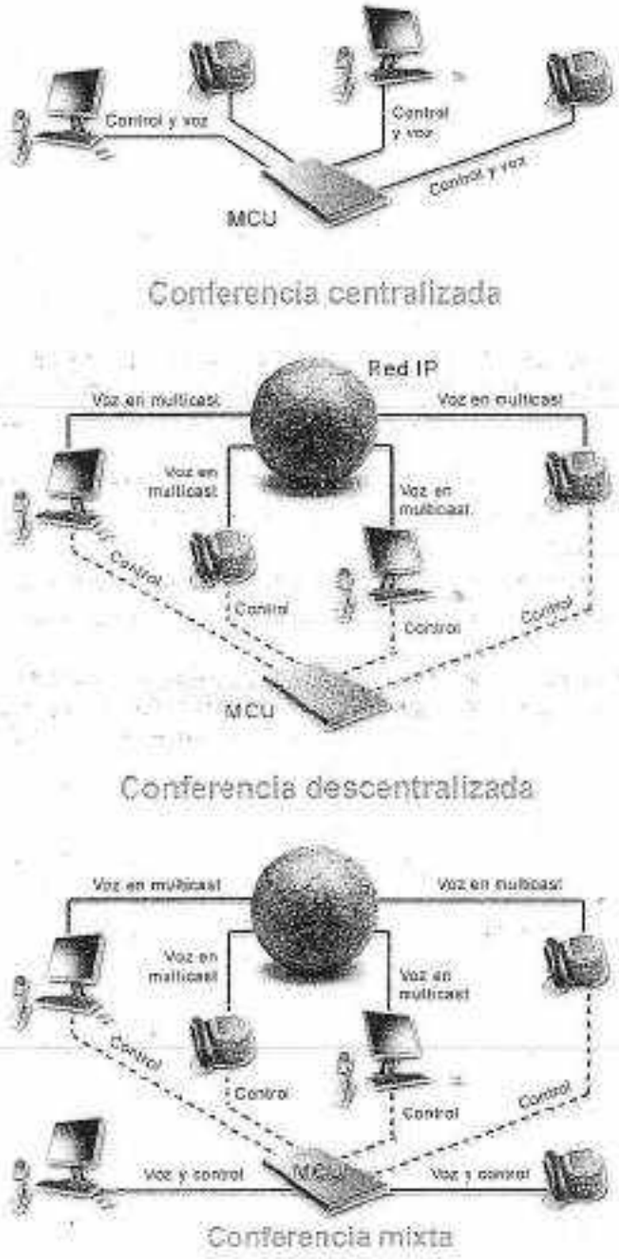

Figura 6.3. Configuraciones de multiconferencia

Aunque lo normal es que estas dos unidades formen parte de un mismo equipo, también podrían estar integradas en equipos distintos. En cualquier caso, la comunicación entre MP y MC no está estandarizada, por lo que ambos tendrían que ser del mismo fabricante.

El MC puede gestionar tres tipos de configuraciones de multiconferencias:

- **Centralizada**. En este caso, cada terminal se comunica con el MCU, quien se encarga, no sólo de controlar la comunicación, sino de distribuir la información multimedia a cada uno de ellos de forma independiente (punto a punto).
- **Descentralizada**. En esta modalidad, cada terminal intercambia la señalización de control con el MC en una comunicación punto a punto, mientras que la información multimedia se intercambia directamente entre los distintos terminales participantes a través de *multicast*. En esta configuración, cada terminal recibe múltiples flujos de audio, vídeo o datos y es responsable de procesarlos. Mediante el canal de control, cada terminal le indica al MC cuántos flujos simultáneos de información multimedia puede manejar.
- **Mixta**. En esta configuración, algunos participantes utilizan la modalidad centralizada (*unicast*) mientras que otros lo hacen de forma descentralizada (*multicast*).

Aunque la modalidad descentralizada supone un ahorro de ancho de banda al utilizar *multicast*, tiene el inconveniente de requerir terminales con mayor capacidad de proceso al tener que decodificar, mezclar y conmutar los distintos flujos de información multimedia que reciben. Por otro lado, los *routers* y *switches* de la red tienen que poder manejar la modalidad *multicast*, cosa que no ocurre en todos los casos. La modalidad centralizada no requiere ninguna capacidad de red especial y admite todo tipo de terminales. Una alternativa intermedia sería que el MCU recibiera la información multimedia punto a punto y transmitiera un flujo *multicast*.

Terminales

Los terminales son los dispositivos finales que utilizan los usuarios de la comunicación multimedia de voz, vídeo o datos. Para asegurarse de que todos los terminales pueden interoperar, H.323 define que todos tienen que admitir, como mínimo, el formato de voz con el codec G.711 (ley A y ley μ). De esta forma, en el caso de que no tengan ningún otro codec en común, al menos, podrán intercomunicarse utilizando G.711, aunque sea a base de utilizar mucho ancho de banda. Por otro lado, el soporte de vídeo y datos es opcional para H.323, pero si lo admite, al menos debe disponer del codec de vídeo H.261 (QCIF, *Quarter Common Intermediate Format*).

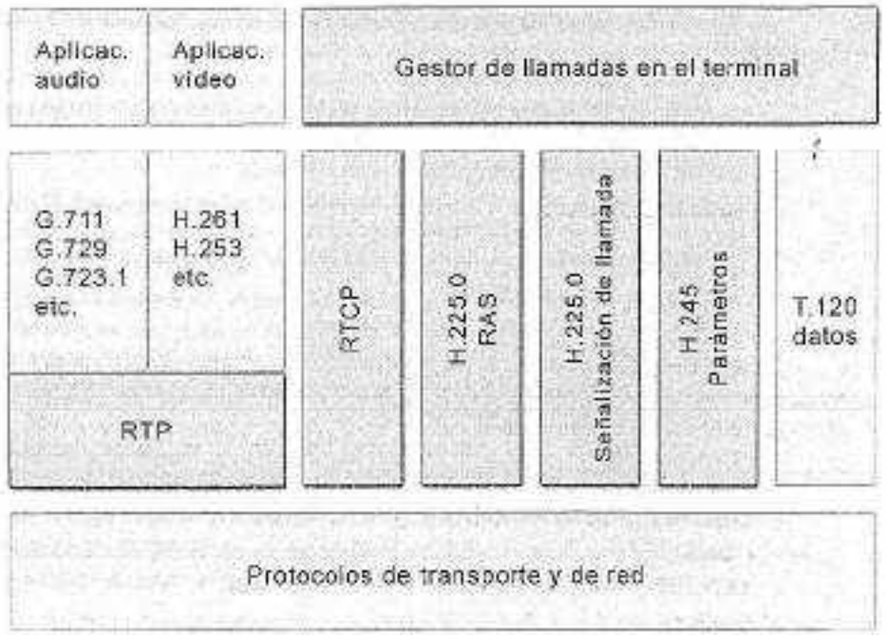

Figura 6.4. Conjunto de protocolos H.323 desde el punto de vista del terminal

Por último, todos los terminales deben admitir los siguientes componentes:

- H.245 para negociar el uso del canal y las características de la comunicación.
- H.225.0 (Q.931) para la señalización y configuración de la llamada.
- H.225.0 (RAS) para comunicarse con el *gatekeeper*.
- RTP/RTCP para el envío de la información multimedia en tiempo real.
- Codecs opcionales de audio y vídeo.
- T.120 para poder utilizar aplicaciones interactivas (como la pizarra, por ejemplo)

EL PROCESO DE SEÑALIZACIÓN

El proceso de la señalización H.323 contempla tanto la señalización necesaria para que el *gatekeeper* pueda controlar los equipos de su zona, como el proceso de intercambio de información necesario para poder establecer y finalizar una comunicación entre equipos finales con distintas características y de distintos fabricantes.

En general, los pasos del proceso de señalización de una llamada H.323 son los siguientes:

- **Gestión con el *gatekeeper***. Se utiliza el protocolo RAS (H.225.0) para que los equipos finales de una red H.323 puedan registrarse en su *gatekeeper*, así como realizar cualquier otra comunicación con el mismo.
- **Establecer la conexión.** Cuando un equipo final desea comunicarse con otro, el primer paso es establecer contacto directo con el mismo. Esto se lleva a cabo con el protocolo conocido como de *Señalización de llamada* (H.225.0).
- **Negociar los parámetros**. Una vez que los equipos participantes se han puesto en contacto, tienen que negociar las condiciones en las que se va a llevar a cabo el intercambio de información multimedia. Este proceso de negociación se lleva a cabo con el protocolo H.245. En esta negociación se establece, por ejemplo, el codec o el tamaño de trama a utilizar.
- **Conversación**. Una vez aclaradas las formas, ya puede dar comienzo la conversación. H.323 tiene establecido que el intercambio de la información multimedia de la conversación se lleve a cabo con el protocolo RTP/RTCP.
- **Terminar**. Por último, cuando un participante da por terminada la conversación, se utiliza de nuevo el protocolo H.225.0 para liberar los recursos utilizados.

Figura 6.5. Proceso de señalización H.323

Como se puede ver, la señalización de H.323 se basa en el empleo de tres protocolos: RAS (H.225.0, en el caso de que se utilice *gatekeeper*), señalización de llamada (H.225.0) y negociación de los parámetros (H.245). Estos protocolos se basan en

el intercambio de mensajes entre los equipos participantes. Aunque existe un cierto orden secuencial en la utilización de cada uno de los protocolos anteriores, lo cierto es que el intercambio de sus mensajes, no sólo son independientes, sino que podría, en un momento dado, realizarse de forma simultánea.

Cada entidad de la red H.323 dispone de una dirección de red que la identifica. En un entorno IP, la dirección de red es una dirección IP. Si la red H.323 dispusiera de un servidor DNS, las direcciones de red podrían especificarse en el formato URL (de acuerdo con las recomendaciones RFC1738 y RFC2396). Por ejemplo, podrían utilizarse direcciones con el formato *ras://gateway@dominio*.

En cuanto a los números de puerto utilizados, éstos los suelen asignar las aplicaciones de forma aleatoria, salvo los relativos a la señalización de llamada (puerto 1720) y a los mensajes RAS: *Descubrir el gatekeeper* (puerto 1718), *Registro* (puerto 1719) y *Estado* (puerto 1718). Estos números de puertos están registrados en IANA (*Internet Assigned Numbers Authority*, 'Autoridad de asignación de números de Internet'). Adicionalmente, los equipos de una red H.323 pueden utilizar alias que son gestionados por el *gatekeeper*. El alias podría ser, por ejemplo, el número telefónico asociado al terminal.

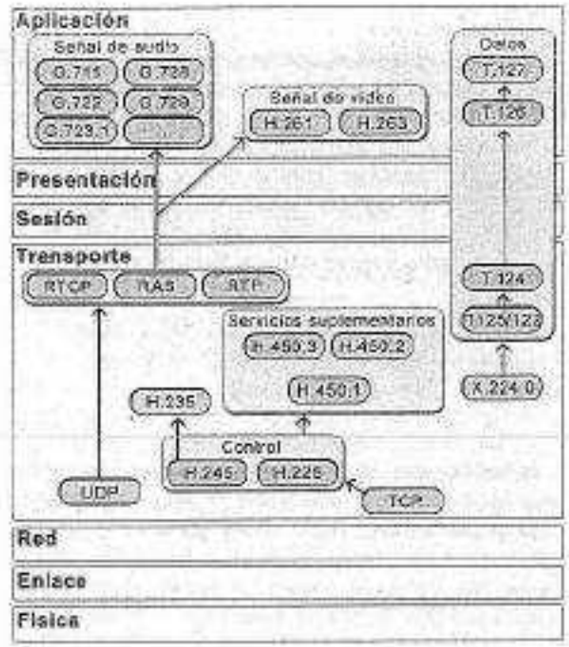

Figura 6.6. H.323 en relación con el modelo OSI

LA COMUNICACIÓN CON EL *GATEKEEPER*. RAS

La señalización RAS (*Registration, Admission and Status*, 'Registro, admisión y estado') viene recogida en la recomendación H.225.0 (junto con la señalización de llamada que se verá más adelante). Como se ha visto anteriormente, esta señalización se utiliza exclusivamente en las comunicaciones de gestión y control entre el *gatekeeper* y los equipos finales de su zona. Esto quiere decir que las redes H.323 que no dispongan de *gatekeeper* no necesitan utilizar este protocolo.

Debido a que la señalización RAS se basa en el envío de mensajes que no suelen tener gran longitud se utilizan paquetes UDP.

Las funciones incluidas en la señalización RAS son las siguientes:

- **Descubrir el *gatekeeper*** (*Gatekeeper discovery*). A veces ocurre que una determinada red dispone de más de un *gatekeeper* con el objeto de tener un reparto de carga o un *gatekeeper* de respaldo. En estos casos, cuando un equipo final (terminal, *gateway* o MCU) va a registrarse puede no saber exactamente el *gatekeeper* que le corresponde. Para descubrirlo bastará con que lance esta pregunta (conocida como GRQ, *Gatekeeper-Request*, 'Solicitud de *gatekeeper*') a las direcciones donde supuestamente se encuentran los *gatekeeper* de la red o a la dirección *multicast* de descubrimiento de *gatekeeper*: 224.0.1.41:1718. La respuesta a estas preguntas será el mensaje GRJ (*Gatekeeper-Reject*, 'Rechazo de *gatekeeper*') o GCF (*Gatekeeper-Confirm*, 'Confirmación de *gatekeeper*'). Hay que tener en cuenta que un equipo final no puede tener asignado más de un *gatekeeper* simultáneamente.
- **Registro** (*Registration*). Permite que cualquier equipo final se registre en un *gatekeeper*. A partir de ese momento pasará a formar parte de la zona de dicho *gatekeeper*. Por cierto, cuando un *gatekeeper* responde GCF a un equipo final no le está indicando que ya está registrado con él, sino, simplemente, que él es el *gatekeeper* que le corresponde (podría estar ya registrado o no). El proceso de registro comienza con el mensaje RRQ (*Registration-Request*, 'Solicitud de registro') dirigido a la dirección IP del *gatekeeper* correspondiente (al número de puerto 1719). En este mensaje, el equipo final le incluye su dirección IP para las comunicaciones RAS y para la señalización de llamada, así como su alias. El *gatekeeper* puede rechazar el registro con un mensaje RRJ (*Registration-Reject*, 'Rechazo del registro'), indicando el motivo, o puede confirmar el registro con un mensaje RCF (*Registration-Confirmation*, 'Confirmación del registro'). Por cierto, si el equipo final no especificaba un alias, el *gatekeeper* podría asignarle uno.
- **Baja** (*Unregistration*). Permite que cualquier equipo final deje de estar bajo el control de un *gatekeeper* particular. A partir de ese momento, el equipo final dejará de formar parte de dicha zona. Para hacer esto el equipo final debe enviar el mensaje URQ (*Unregistration-Request*, 'Solicitud de baja') al *gatekeeper*, a lo que el *gatekeeper* responderá con el mensaje UCF (*Unregistration-Confirmation*, 'Confirmación de baja'). El *gatekeeper* también puede decidir la cancelación del registro de un equipo final enviándole un mensaje URQ.
- **Admisión** (*Admission*). Este es el mensaje que utiliza un equipo final para solicitar de su *gatekeeper* su participación en una llamada (de hecho, creo que a este mensaje le vendría mejor el nombre *participación* que el de *admisión*). El equipo final debe incluir en el mensaje de admisión (conocido como ARQ, *Admission-Request*, 'Solicitud de admisión') el tipo de llamada de que se trata (entre dos o multiconferencia), su identificador, el ancho de banda que necesita (en unidades de 100 bits por segundo), un identificador de llamada, así como información (alias o direcciones) sobre el resto de participantes. El *gatekeeper* responderá afirmativamente con un mensaje ACF (*Admission-Confirm*, 'Confirmación de admisión') o negativamente con un mensaje ARJ (*Admission-Reject*, 'Rechazo de admisión'). En el primer caso, el *gatekeeper* confirmará, entre otras cosas, el ancho de banda disponible, que podría ser menor del solicitado. Otra información interesante que se incluye en estos mensajes es si la señalización de llamada se hará directamente entre los equipos finales o si, por el contrario, el *gatekeeper* hará de intermediario (parámetro *CallModel*). En este sentido, el terminal podría sugerir una modalidad en su mensaje ARQ, a lo que el *gatekeeper* responderá en su mensaje ACF. Por cierto, un *gatekeeper* podría incluir en su mensaje ACF el parámetro *preGrantedARQ* (otorgada admisión previa) para indicarle al equipo final que no es necesario que le pida permiso (que le envíe mensajes ARQ) en sus comunicaciones futuras.

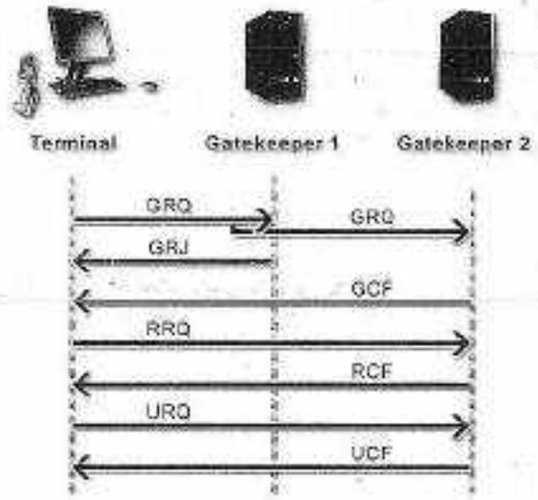

Figura 6.7. Descubrimiento y registro en el gatekeeper

- **Cambio de ancho de banda** (*Bandwidth change*). Durante el tiempo que dura una llamada, un equipo final puede darse cuenta de que necesita un ancho de banda mayor o menor del que tiene concedido por el *gatekeeper*. Si de lo que se trata es de reducir el ancho de banda, puede hacerlo directamente sin más (aunque puede informar al *gatekeeper* de esta reducción); pero si lo que necesita es aumentar el ancho de banda, deberá solicitar un permiso previo al *gatekeeper*. El mensaje de solicitud se conoce como BRQ (*Bandwidth-Request*, 'Solicitud de ancho de banda'), donde se indicará el nuevo ancho de banda necesario. El *gatekeeper* puede aprobar esta petición con un mensaje BCF (*Bandwidth-Confirm*, 'Confirmación de ancho de banda') o denegarla con un mensaje BRJ (*Bandwidth-Reject*, 'Rechazo de ancho de banda'). El *gatekeeper* también podría iniciar una petición de cambio de ancho de banda (por ejemplo, por saturación de la red) enviándole un mensaje BRQ al equipo final al que sólo cabe la respuesta BCF. Por cierto, los cambios de ancho de banda afectan a la señalización H.245 (por ejemplo, puede que haya que elegir otro codec).
- **Localización** (*Endpoint location*). Se trata de una función por la que un equipo final puede solicitar la dirección real de otro del que sólo dispone del alias. Para ello envía un mensaje LRQ (*Location-Request*, 'Solicitud de identificación') a la dirección de un *gatekeeper* particular o a la dirección *multicast* de los *gatekeeper* (224.0.1.41). Este mensaje LRQ puede ser utilizado también para que un *gatekeeper* le envíe la pregunta a otro *gatekeeper*. La respuesta puede ser un LCF (*Location-Confirm*, 'Confirmación de identificación') donde se indica la dirección solicitada o un LRJ (*Location-Reject*, 'Rechazo de identificación') para indicar que no se conoce dicha información. Si un *gatekeeper* recibe la pregunta vía *multicast* no generará una respuesta LRJ. La respuesta LRJ lo que viene a indicar es que dicho equipo final no está registrado en el *gatekeeper*.
- **Desconexión** (*Disengage*). Una vez que termina la comunicación, los distintos equipos finales deben comunicarle este estado a su *gatekeeper* correspondiente enviándole un mensaje DRQ (*Disengage-Request*, Solicitud de desconexión). Entre otras informaciones, el mensaje contendrá el identificador de la llamada y una razón por la que se desconecta (por ejemplo, llamada terminada con normalidad). El *gatekeeper* responderá a este mensaje DRQ con un mensaje de confirmación DCF (*Disengage-Confirm*, Confirmación de desconexión). No obstante, bajo ciertas circunstancias (por ejemplo, resulta que el equipo final no está registrado en dicho *gatekeeper*), también podría responder con un mensaje DRJ (*Disengage-Reject*, Rechazo de desconexión). Si es el *gatekeeper* el que decide dar por terminada la conexión, le enviará un mensaje DRQ al equipo final correspondiente; quien deberá cerrar la sesión siguiendo el protocolo H.245 y H.225.0. Una vez terminada la sesión se lo confirmará al *gatekeeper* enviándole un mensaje DCF.
- **Estado** (*Status*). Por razones de control, puede resultar conveniente que un *gatekeeper* conozca el estado de los equipos finales de su zona y de las llamadas en curso. El *gatekeeper* puede obtener esta información de un equipo final de dos formas: enviándole un mensaje IRQ (*Information-Request*, 'Solicitud de información') o indicándole en un mensaje ACF que le envíe periódicamente dicha información (cada cierto número de segundos definido en el parámetro *irrFrequency*). El equipo final responde a la petición de información con un mensaje IRR (*Information-Request-Response*, 'Respuesta a la solicitud de información') donde le informará de sus llamadas en curso (identificador de llamada, si es punto a punto o multiconferencia, el ancho de banda, dirección RTP, etc.). Por cierto, entre el *gatekeeper* y el equipo final se acuerda si el *gatekeeper* responderá con un mensaje de confirmación a cada mensajes IRR que reciba (mensajes IACK, *Info-Request-Ack*, o INAK, *Info-Request-Nak*).
- **Disponibilidad** (*Resource availability*). Este mensaje lo utilizan los *gateways* para informarle al *gatekeeper* de los recursos que tiene actualmente disponibles. Esta información se envía en un mensaje RAI (*Resource-Available-Indicate*, 'Indicación de recursos disponibles'), al que el *gatekeeper* confirmará la recepción con un mensaje RAC (*Resource-Available-Confirm*, 'Confirmación de recursos disponibles'). En el mensaje RAI se incluye la disponibilidad actual del *gateway* en cuanto al número de llamadas y ancho de banda disponible para cada uno de los protocolos que admite. Si el *gateway* estuviera en los límites de sus recursos incluiría el parámetro *AlmostOutOfResources* (Casi sin recursos). Las circunstancias bajo las que el *gateway* envía este mensaje de alarma dependen de cada fabricante.
- **No regulada** (*Non-standard*). Se trata de un mecanismo por el cual se puede intercambiar cualquier información no regulada (propietaria) entre cualquier equipo final y el *gatekeeper*. Para ello se utilizan los mensajes SCI (*Service-Control-Indication*, 'Indicación de control') y SCR (*Service-Control-Response*, 'Respuesta de control'). Estos mensajes se incluyeron en la versión 4 de H.323.
- **Solicitud en curso** (*Request in Progress*). Muchos de los mensajes RAS tienen establecidos tiempos de espera. Esto quiere decir que, de no recibirse una respuesta dentro de este tiempo límite, se vuelve a enviar la solicitud. No obstante, puede ocurrir que un equipo final prevea que va a tardar algo más de lo normal en responder a una solicitud. En este caso, envía un mensaje RIP (*Request-In-Progress*, 'Solicitud en curso') para indicar que le dé algo más de tiempo.

ESTABLECER LA CONEXIÓN. H.225.0

La señalización de llamada es el procedimiento que utilizan los equipos finales para posibilitar el establecimiento y la terminación de las llamadas. Estos mensajes de señalización se intercambian entre los equipos finales directamente, por lo que el equipo llamante debe conocer previamente la dirección del equipo llamado. Esta dirección puede venir definida por el número IP o por un alias. Un número telefónico es un tipo particular de alias. Si el terminal sólo dispone del alias, puede utilizar los servicios del *gatekeeper* para averiguar la dirección IP correspondiente.

La señalización de llamada se basa en el intercambio de mensajes entre los equipos participantes. Para la definición de estos mensajes, el protocolo H.225.0 se ha basado en el formato utilizado por Q.931 (utilizado por la RDSI en la señalización del interfaz usuario-red de RDSI). Por este motivo, a la señalización de llamada también se la conoce como señalización Q.931. No obstante, H.225.0 no utiliza todos los mensajes de Q.931, sino sólo aquellos que tienen sentido dentro de la arquitectura H.323. Por cierto, también utiliza un mensaje de Q.932.

En las primeras versiones de H.323 se contemplaba que la señalización de llamada debía hacerse sobre TCP, no obstante, con la idea de agilizar este intercambio de información, la versión 4 de H.323 ya contempla que pueda realizarse tanto sobre TCP como UDP.

Los mensajes de señalización de llamada definidos en H.225.0 son los siguientes:

- **Establecimiento** (*Setup*). Se trata del primer mensaje que el equipo final llamante envía al llamado para poder establecer la comunicación. Este mensaje incluye unos elementos de información obligatorios (por ejemplo, el identificador de llamada, el tipo de llamada, el identificador de conferencia o información sobre el equipo final de origen) y otros opcionales (por ejemplo, el alias de origen, alias de destino o la dirección H.245 para los mensajes subsecuentes). Si no se utiliza un *gatekeeper*, el mensaje de establecimiento lo enviará directamente el equipo que origina la llamada al destinatario. Si se utiliza un *gatekeeper*, el equipo de origen tiene que obtener la autorización del mismo (mensaje *Admisión* de RAS) antes de enviar este mensaje.
- **Procediendo** (*Call-Proceeding*). Se trata de un mensaje que envía, opcionalmente, el equipo receptor del mensaje de establecimiento para indicar que está procediendo a llevar a cabo el establecimiento de la llamada. Este mensaje se envía antes del mensaje *Conectar (Connect)*.
- **Alerta** (*Alerting*). Este mensaje lo envía el equipo final que recibe la llamada para indicar que está avisando al usuario para que atienda la llamada. Es lo equivalente al tono de llamada en la telefonía tradicional.
- **Progreso** (*Progress*). Se trata de un mensaje que puede enviar un *gateway* para indicar que se está procediendo a establecer la conexión. Este mensaje se utiliza, especialmente, en el caso de interconexión con la red telefónica tradicional.
- **Conectar** (*Connect*). Este mensaje lo envía el equipo llamado para indicar que su usuario ha aceptado la llamada.
- **Liberación** (*Release Complete*). Este mensaje se utiliza para informar al otro participante de la comunicación que la llamada se ha dado por terminada. Opcionalmente se puede incluir en el mensaje la causa de la terminación de la llamada.
- **Facilidad** (*Facility*). Este mensaje se utiliza para solicitar un servicio suplementario (por ejemplo, redirección de llamada). Este mensaje lo toma H.225.0 de la recomendación Q.932 y H.450.1 de la UIT-T.

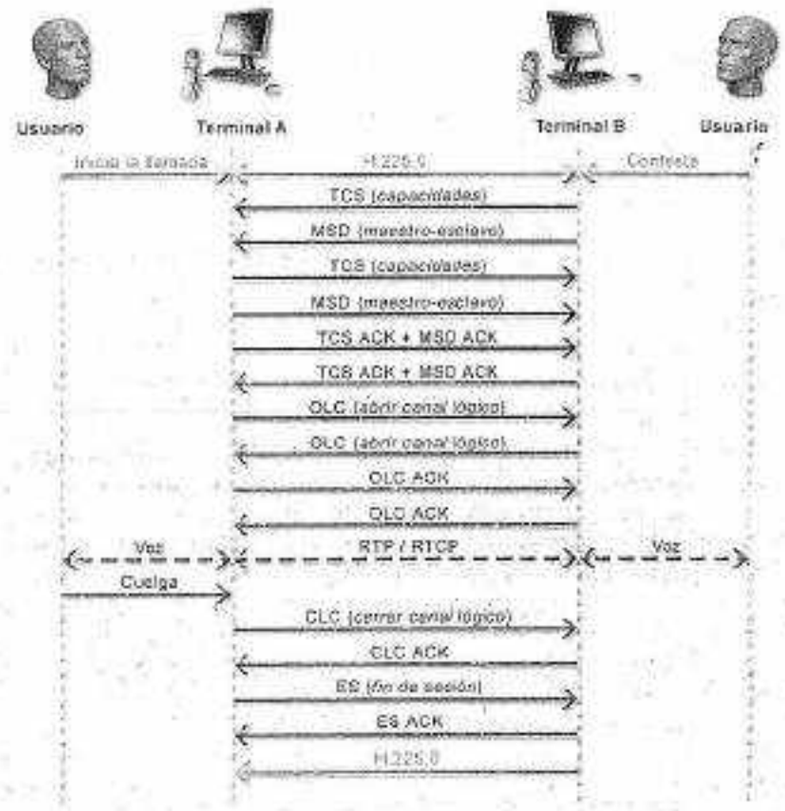

Figura 6.8. Establecimiento de la conexión con H.225.0

NEGOCIAR LOS PARÁMETROS. H.245

Si el equipo llamante ha realizado la llamada correspondiente (mensaje *Establecimiento* de H.225.0) y el equipo llamado ha respondido positivamente (mensaje *Conectar* de H.225.0), lo que viene a continuación es negociar los términos en los que se va a producir el intercambio de información multimedia: codec a utilizar, tamaño de trama, tipo de medio (voz, vídeo o datos), selección de puertos, etc. Esta negociación entre equipos finales se lleva a cabo utilizando el protocolo H.245.

Aunque en un proceso secuencial se podría esperar que la negociación de las condiciones comenzará justo a continuación de que el equipo llamado haya aceptado la

llamada (mensaje *Conectar*), lo cierto es que el trabajo de H.245 es puramente administrativo, por lo que cuanto antes se haga, mejor. En la práctica, los mensajes H.245 pueden empezar a intercambiarse justo a continuación del mensaje *Establecimiento*. Esto hace que la llamada pueda comenzar sin demora inmediatamente después de que el usuario llamado acepte la llamada.

Por otro lado, H.245 no se dedica exclusivamente a la VoIP, sino que sus funciones se utilizan en un gran número de aplicaciones. Esto hace que el número de mensajes establecidos por esta recomendación sea bastante elevado. En general, estos mensajes pueden encuadrarse dentro de los siguientes grupos:

- **Peticiones** (*Requests*). Se trata de mensajes que solicitan que el receptor realice una cierta acción y envíe un mensaje de respuesta inmediatamente.
- **Respuestas** (*Responses*). Estos mensajes se envían como respuesta a un mensaje de petición.
- **Órdenes** (*Commands*). Estos mensajes se envían para que el receptor realice una acción concreta, sin que sea necesario un mensaje de respuesta.
- **Indicaciones** (*Indications*). Se trata de mensajes de naturaleza exclusivamente informativa. En este caso no se pide que se realice ninguna acción ni se espera ninguna respuesta.

Intercambio de capacidades

La negociación de las características de la información multimedia que se va a intercambiar forma parte de un procedimiento que se conoce como *Intercambio de capacidades* (*Capabilities Exchange*). Esto consiste en que cada equipo final participante envía un mensaje TCS (*TerminalCapabilitySet*, 'Conjunto de capacidades del terminal') exponiendo sus propias capacidades y preferencias (codecs que puede manejar, preferencia de codecs a utilizar, combinaciones particulares de voz y vídeo, etc.). De la misma forma, cada equipo toma nota de las capacidades del otro (u otros) para enviar su información con las características necesarias para que pueda ser entendida por el destinatario.

El mensaje TCS es de tipo *Petición* (*Request*), por lo que tiene contemplado los mensajes de respuesta *TerminalCapbilitySetAck* ('Aceptación del conjunto de capacidades') para dar por buena estas capacidades, y *TerminalCapabilitySetReject* ('Rechazo del conjunto de capacidades') si es que se detecta algún tipo de problema. Si, por algún motivo, un equipo final necesita conocer una característica particular de otro o necesita que le repita todas sus capacidades, puede enviarle el mensaje *SendTerminalCapabilitySet* ('Enviar el conjunto de capacidades'), a lo que se le responderá con el mensaje TCS.

Determinación maestro-esclavo

En la comunicación H.245 entre terminales distantes pueden surgir situaciones de conflicto. Por ejemplo, más de un equipo final de una multiconferencia puede disponer de un MC que pretenda tener el control, o dos equipos finales pueden intentar comunicar a la vez. Para estos casos, H.245 dispone de un procedimiento que permite decidir el equipo que tiene el mando. Este procedimiento se denomina *Determinación maestro-esclavo*.

Para determinar el terminal maestro y el esclavo se utilizan dos datos: el tipo de terminal y un número aleatorio entre 1 y 16.777.215.

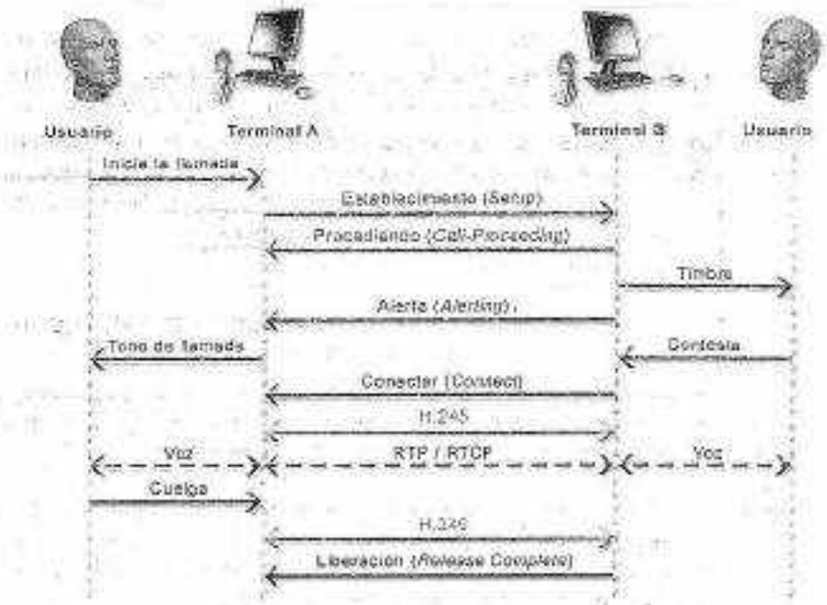

Figura 6.9. Negociar los parámetros con H.245

H.323 le asigna un número a cada tipo de equipo, conocido como valor de terminal. Por ejemplo, un equipo que no dispone de MC tiene un valor de terminal 60, un MCU que admite multiconferencias de voz, vídeo y datos tiene un valor de 190 y un MCU que está gestionando actualmente una multiconferencia tiene el valor 240 (el más alto posible).

El procedimiento de determinación maestro-esclavo le asigna el papel de maestro al terminal cuyo valor de terminal sea mayor. En caso de empate se escogería como

maestro el terminal que tenga un mayor valor aleatorio. Los mensajes que intervienen en esta determinación se conocen como MSD (*Master-Slave Determination*, 'Determinación maestro-esclavo').

MENSAJE	FUNCIÓN	POSIBLE RESPUESTA
Determinación maestro-esclavo (Master-Slave Determination)	*Determina el terminal que actuará como maestro y que lo hará de esclavo.*	*Acknowledge, Reject, Release*
Conjunto de capacidades del terminal (Terminal Capability Set)	*Contiene información sobre las capacidades de envío y recepción de información multimedia.*	*Acknowledge, Reject, Release*
Abrir canal lógico (Open Logical Channel)	*Abre un canal lógico para el transporte de información multimedia o datos.*	*Acknowledge, Reject, Confirm*
Cerrar canal lógico (Close Logical Channel)	*Cierra el canal lógico establecido entre dos equipos finales.*	*Acknowledge*
Modo petición (Request Mode)	*Utilizado por un terminal receptor para solicitarle al transmisor que le transmita en un modo particular.*	*Acknowledge, Reject, Release*
Enviar conjunto de capacidades de terminal (Send Terminal Capability Set)	*Le pide al terminal del otro extremo que le envíe sus capacidades de envío y recepción.*	*Terminal Capability Set*
Orden de finalización de sesión (End Session Command)	*Indica la terminación de la sesión H.245.*	

Tabla 6.2. Mensajes del protocolo H.245

Abrir el canal lógico

H.245 utiliza el término de canal lógico para hacer referencia al flujo de información multimedia entre los equipos finales. Un canal lógico es un enlace unidireccional entre dos equipos finales. Por tanto, una comunicación telefónica bidireccional entre dos participantes utilizará dos canales lógicos. Esta separación permite, por ejemplo, que cada canal lógico utilice un codec distinto.

Desde el punto de vista de una red IP, un canal lógico viene a ser una dirección IP y un número de puerto que están asociados e identificados con un número de canal lógico.

Todos los equipos finales tiene establecido un canal lógico de control por el que reciben los mensajes de control H.245 (el canal número 0). Cuando un equipo va a transmitir un flujo multimedia, previamente envía un mensaje al destinatario para abrir el canal lógico correspondiente (mensaje OLC, *OpenLogicalChannel*, 'Abrir canal lógico').

En este mensaje le indica el número de canal lógico y el tipo de información que va a transmitir. El receptor responde con un mensaje de aceptación (mensaje *OpenLogicalChannelAck*, 'Aceptación para abrir canal lógico') donde incluye el número de puerto en el que desea recibir este flujo de información multimedia.

Si, por algún motivo, el destinatario no desea abrir este canal debe responder con un mensaje de rechazo (mensaje *OpenLogicalChannelReject*, 'Rechazo para abrir canal lógico'). Si el equipo final llamado desea enviar también información multimedia, necesita abrir otro canal lógico distinto en la dirección contraria. También existe la posibilidad de abrir ambos canales con un solo mensaje de apertura y otro de aceptación (canal bidireccional). En este caso, en el mensaje de apertura se incluirán los detalles de ambos canales.

Una vez terminado el envío de datos, el terminal que abrió el canal lógico debe cerrarlo con el envío del mensaje CLC (*CloseLogicalChannel*, 'Cerrar canal lógico'). La respuesta será un mensaje de aceptación de cierre (*CloseLogicalChannelAck*, 'Aceptación para cerrar el canal lógico'). Si el que desea cerrar el canal es el terminal receptor, debe enviar un mensaje de solicitud de cierre (mensaje *RequestChannelClose*, 'Solicitud de cierre del canal') a lo que el emisor responderá con un mensaje de confirmación.

Una vez cerrados los canales de una sesión, cualquier equipo final puede proceder a cerrar la sesión con el envío de un mensaje *EndSession* ('Fin de sesión') a lo que el otro equipo responderá con otro mensaje *EndSession*. Este será el último mensaje H.245 de la sesión.

Procedimiento de conexión rápida

Si revisamos el procedimiento de establecimiento de la comunicación descrito anteriormente, nos damos cuenta de que es un proceso largo, lleno de mensajes de ida y vuelta, con todos los inconvenientes que eso supone. Para acortar el tiempo necesario para establecer la conexión, H.323 se ha definido un procedimiento abreviado conocido como de conexión rápida (*Fast-Connect procedure*).

El procedimiento de conexión rápida consiste en enviar un mensaje *OpenLogicalChannel* que contenga toda la información necesaria, incluidos los parámetros del canal inverso (si es que la comunicación va a ser bidireccional). De la misma forma, el destinatario responde con todos los elementos necesarios incluidos en un mensaje. Si el destinatario no responde con toda la información, se asume que no admite el procedimiento rápido y se pasa a utilizar el procedimiento normal H.245.

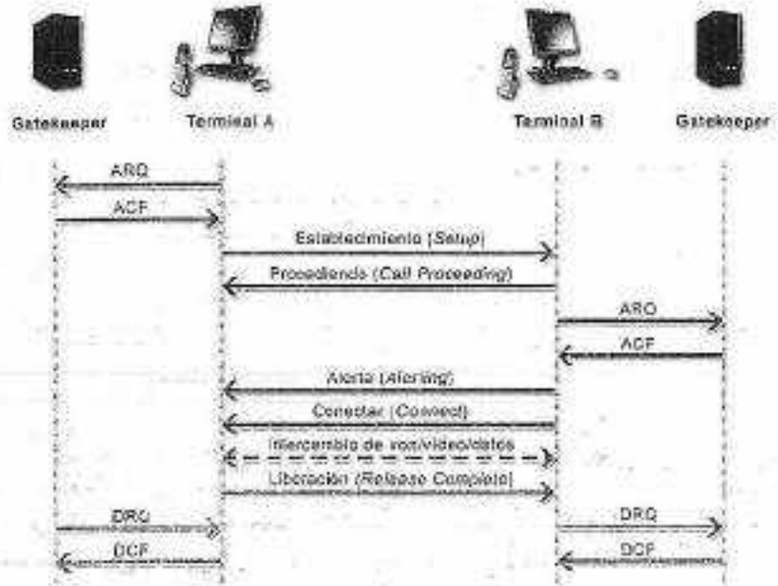

Figura 6.10. Procedimiento rápido de conexión.

MULTICONFERENCIAS CON H.323

Existen dos circunstancias por las que se puede establecer una comunicación en la que intervengan tres o más participantes: una multiconferencia que se establece con la participación de un MCU, o bien, en una conferencia existente de dos participantes se decide espontáneamente ampliarla. La primera de ellas se conoce como conferencia acordada, mientras que la otra recibe el nombre de conferencia *ad-hoc* (a medida). A las multiconferencias *ad-hoc* son a las que estamos acostumbrados en los sistemas de telefonía tradicional.

En las conferencias acordadas, cada participante se conecta a la multiconferencia estableciendo una llamada con el MCU que la controla. El modo en el que se va a intercambiar la información multimedia (centralizado o descentralizado con *multicast*) lo define el MCU mediante el envío de un mensaje a todos los participantes.

En el caso de las conferencias *ad-hoc*, se lleva a cabo utilizando la funcionalidad MC incluida en uno de los terminales participantes o en el *gatekeeper*.

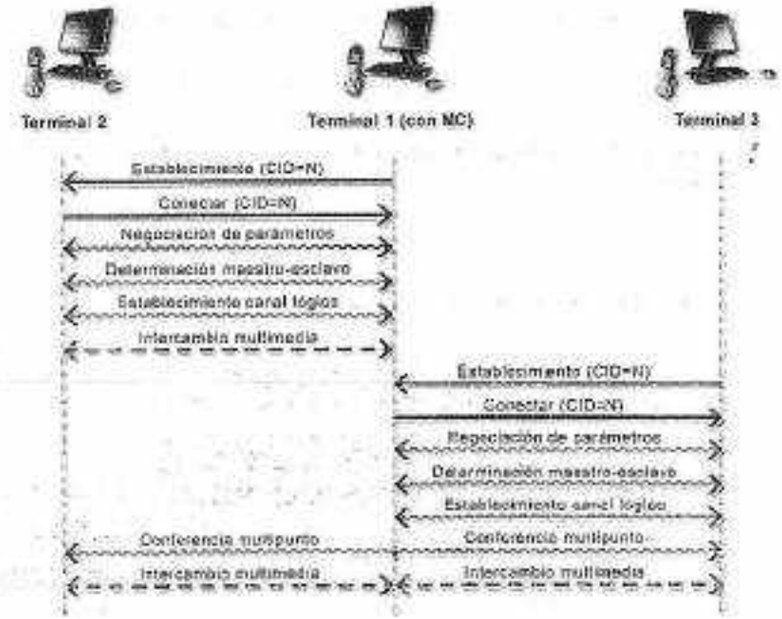

Figura 6.11. Establecimiento de una conferencia a tres (Ad-hoc)

PONERLO TODO JUNTO

Después de haber descrito cada uno de los protocolos que forman parte de una llamada H.323, llega el momento de ponerlo todo junto para recrear cómo sería el proceso completo de establecimiento de una llamada.

Como se ha visto anteriormente, una llamada H.323 puede realizarse directamente entre dos terminales o puede contar con la participación de un *gateway*, de uno o más *gatekeeper* y de una MCU, si se trata de una multiconferencia. Por otro lado, cada *gatekeeper* que intervenga puede estar configurado para limitarse a autorizar la llamada o puede también hacer de intermediario en la señalización de la misma. Quiere esto decir que en los procesos de señalización se pueden dar múltiples combinaciones.

En la idea de darle forma a los distintos conceptos expuestos anteriormente, vamos a ver a continuación algún ejemplo significativo de una conexión H.323. Pero antes, recordar que H.323 contempla los siguientes pasos:

- Establecer la conexión utilizando H.225.0 (Señalización de llamada).

- Negociar los parámetros del intercambio de información multimedia utilizando el protocolo H.245.
- Llevar a cabo la conversación. Cada terminal utilizará el codec y la configuración acordada en la negociación anterior.
- Dar por terminada la comunicación, liberando los recursos implicados (H.245 y H.225.0).

Llamada directa entre terminales

Para este primer ejemplo vamos a suponer que un terminal A realiza una llamada a un terminal B de forma directa; sin que intervenga ningún otro equipo. No hay *gateway*, ni MCU, ni *gatekeeper*.

La llamada comienza cuando el usuario llamante levanta el teléfono IP (software o hardware) y teclea la identificación del destinatario. En ese momento, el terminal A envía al terminal B el mensaje H.225.0 **Establecimiento** (*Setup*). El terminal B responde con el mensaje H.225.0 **Procediendo** (*Call-Proceeding*). A continuación hace sonar el timbre para avisar a su usuario y le envía el mensaje H.225.0 **Alerta** (*Alerting*) al terminal A para indicarle que ya está siendo avisado el destinatario. En este momento, el terminal A puede enviarle un tono de llamada a su usuario para indicarle que le está sonando el teléfono al usuario llamado.

Cuando el destinatario descuelga, el terminal B le envía un mensaje H.225.0 **Conectar** (*Connect*) al terminal A y da comienzo la fase de negociación de parámetros. Ambos terminales envían un mensaje H.245 **TCS** (*TerminalCapabilitySet*, 'Conjunto de capacidades del terminal') y H.245 **MSD** (*Master-Slave Determination*, 'Determinación maestro esclavo'), respondiendo con un mensaje de aceptación para cada uno de los mensajes anteriores (*TCS Ack* y *MSD Ack*).

Una vez que cada terminal conoce las capacidades del otro y que se ha decidido quién manda, se pasa a abrir el canal lógico. Para ello, cada terminal envía un mensaje H.245 **OLC** (*OpenLogicalChannel*, 'Abrir canal lógico') al que el otro terminal responderá con un mensaje de aceptación para abrir el canal lógico (*OLC Ack*). A partir de este momento ambos usuarios pueden mantener su conversación. La voz será digitalizada por su terminal correspondiente y transmitida en tiempo real con el protocolo RTP/RTCP.

Al colgar uno de los usuarios, su terminal envía el mensaje H.245 **CLC** (*CloseLogicalChannel*, 'Cerrar canal lógico'), a lo que el otro le responderá con el mensaje de aceptación de cierre del canal lógico (*CLC Ack*). A continuación, ambos terminales avisarán de que dan por terminada la sesión con el envío de un mensaje H.245 **Fin de sesión** (*EndSession*) y H.225.0 **Liberación** (*ReleaseComplete*).

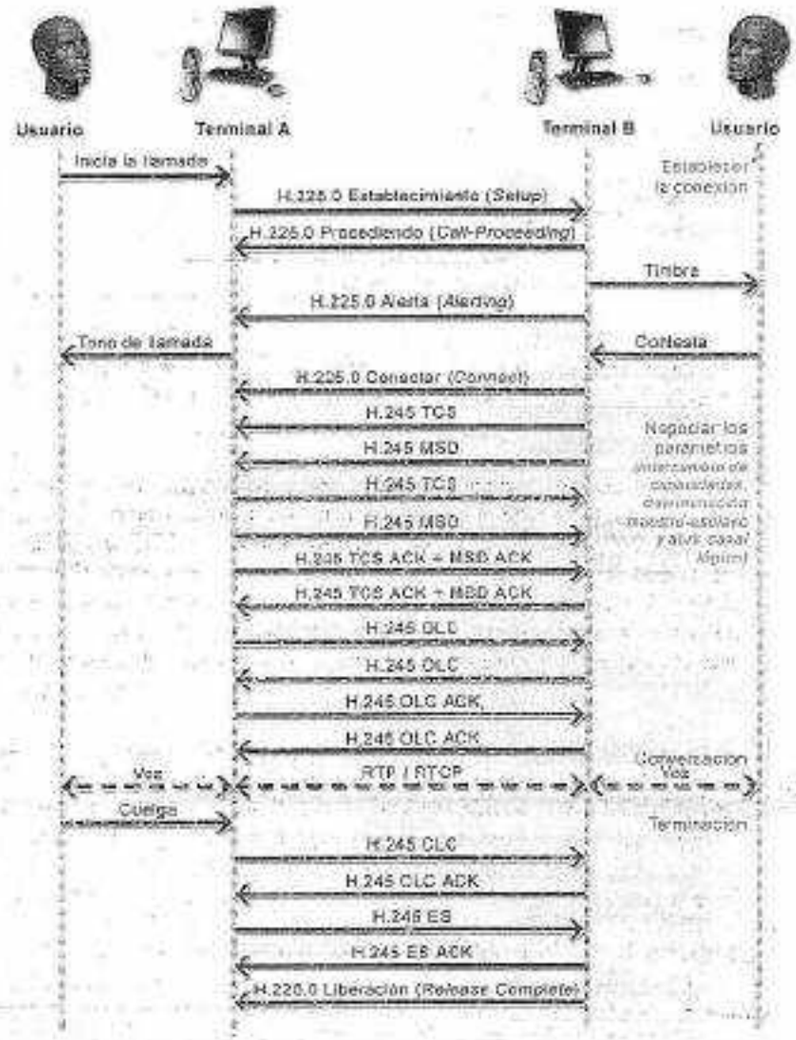

Figura 6.12. Flujo de mensajes de señalización entre dos terminales

Llamada con *gatekeeper*

En este caso, supongamos que tenemos un terminal A dependiente de un *gatekeeper* A, que desea comunicarse con el terminal B dependiente del *gatekeeper* B.

En este caso, antes de establecer la conexión es necesario que cada terminal le pida permiso a su *gatekeeper* correspondiente. Vamos a suponer que ningún terminal tiene otorgada la admisión previa.

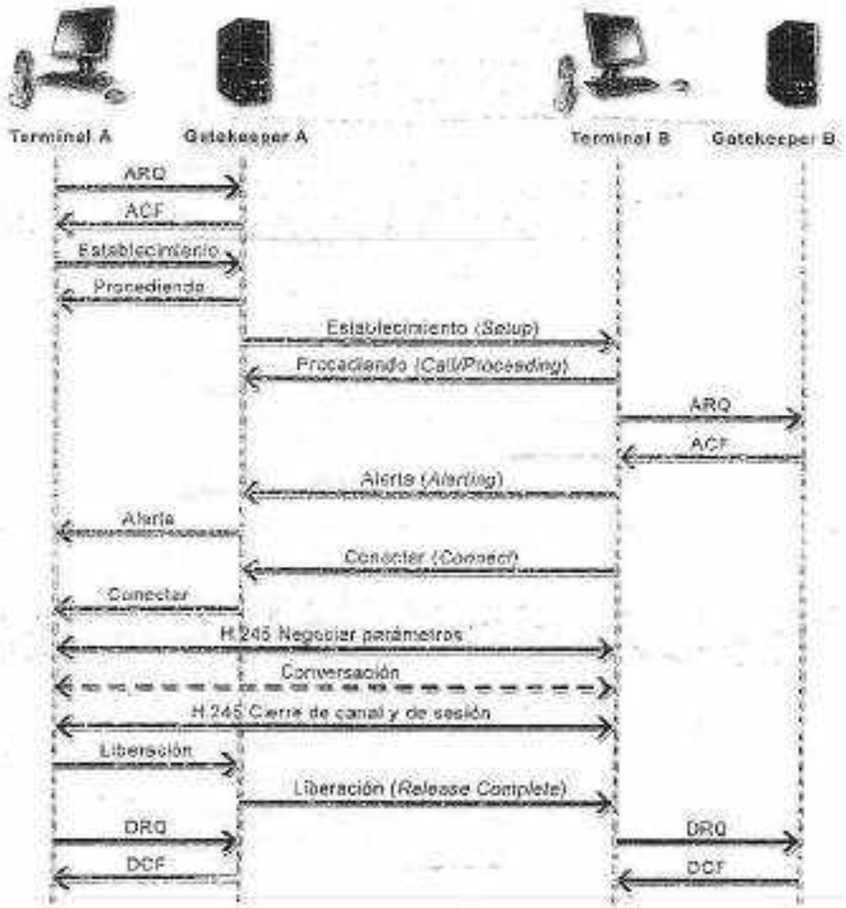

Figura 6.13. Señalización de una llamada con gatekeeper intermediario

Como se ha visto anteriormente, el terminal solicita el permiso de conexión con el mensaje RAS **ARQ** (*Admission-Request*, 'Solicitud de admisión'), a lo que el *gatekeeper* responderá con un mensaje RAS **ACF** (*Admission-Confirm*, 'Confirmación de admisión').

A partir de este momento ya se puede iniciar la señalización de la llamada, pero pueden darse dos circunstancias: que el *gatekeeper* deje a su terminal realizar la señalización directamente o que quiera estar como intermediario. Para este ejemplo podemos suponer que el *gatekeeper* A desea controlarlo todo, mientras que el *gatekeeper* B le deja esta tarea a su equipo final.

En este caso, el terminal A le enviará a su *gatekeeper* el mensaje H.225.0 **Establecimiento** (*Setup*), a lo que el *gatekeeper* le responderá con el mensaje H.225.0 **Procediendo** (*Call-proceeding*). A continuación el *gatekeeper* le enviará el mensaje H.225.0 **Establecimiento** (*Setup*) al terminal B y obtendrá de él el mensaje H.225.0 **Procediendo** (*Call-proceeding*). El terminal B le pedirá permiso a su *gatekeeper* enviándole un mensaje RAS **ARQ**, a lo que éste le responderá con el mensaje RAS **ACF**.

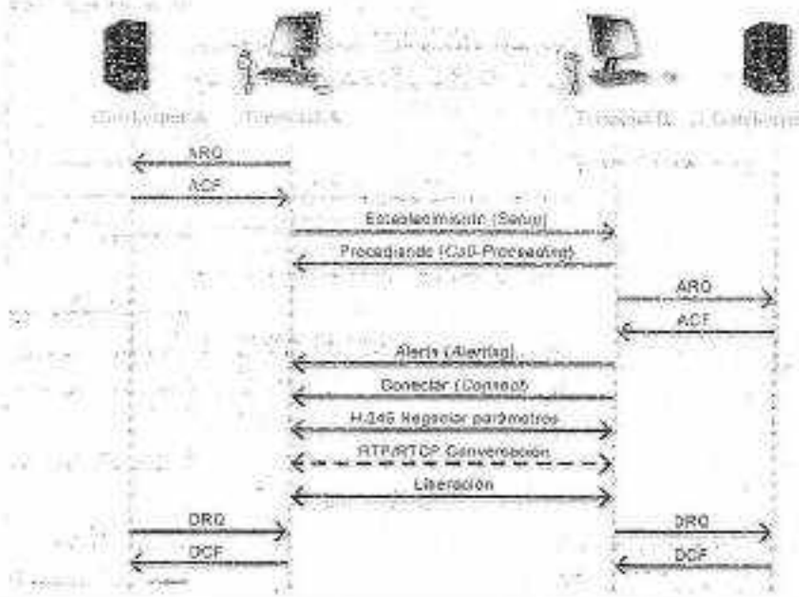

Figura 6.14. Señalización directa entre terminales con gatekeepers

Una vez obtenido el permiso correspondiente, el terminal B le envía el mensaje H.225.0 **Alerta** (*Alerting*) al *gatekeeper* A, quien a su vez le envía un mensaje H.225.0 **Alerta** (*Alerting*) a su terminal A. La llamada se completa enviándole el terminal B un mensaje H.225.0 **Conectar** (*Connect*) al *gatekeeper* A, quien a su vez le envía este mensaje H.225.0 **Conectar** (*Connect*) al terminal A.

A partir de aquí, el *gatekeeper* A deja que ambos terminales hablen directamente. Se establece la negociación H.245 y se procede al intercambio de información multimedia utilizando RTP/RTCP.

Al concluir la comunicación, ambos terminales se intercambian los mensajes H.245 de cierre de canal y de sesión. A continuación, el terminal A le envía el mensaje H.225.0 **Liberación** (*Release Complete*) a su *gatekeeper*, quien a su vez se lo envía al terminal B. Por último, ambos terminales le envían el mensaje RAS **DRQ** (*Disengage-Request*, 'Petición de desconexión') a sus *gatekeeper*, quienes le responden con el mensaje RAS **DCF** (*Disengage-Confirm*, 'Confirmación de desconexión').

Como se ha podido apreciar, mientras que el terminal A ha realizado toda su señalización de llamada a través de su *gatekeeper*, el terminal B lo único que ha tenido que comunicarle a su *gatekeeper* es el comienzo y terminación de la llamada. No obstante, el intercambio de información multimedia se realiza directamente entre los terminales.

Establecimiento de la comunicación con SIP

INTRODUCCIÓN

El IETF creó en 1996 un grupo de trabajo conocido como MMUSIC (*Multiparty Multimedia Session Control*, 'Control de sesión multimedia y multiparticipante') con el objetivo de desarrollar un protocolo que permitiera la comunicación multimedia (audio, vídeo, etc.) entre múltiples usuarios. Ese mismo año propondrían un documento que surgió como resultado de la fusión de dos trabajos: SIP (*Session Initiation Protocol*, 'Protocolo de inicio de sesión'), de Mark Handley y E. Schooler, y SCIP (*Simple Conference Invitation Protocol*, 'Protocolo simple de invitación a conferencia') de Henning Schulzrinne.

A pesar de que el documento final adoptara el nombre de SIP, la gran aportación de este nuevo protocolo vendría de Schulzrinne: utilizar la filosofía del protocolo HTTP usado en las páginas Web. Henning Schulzrinne, profesor asociado del Departamento de Ciencia Informática de la Universidad de Columbia y coautor de los protocolos RTP y RTSP (*Real Time Streaming Protocol*, 'Protocolo de flujo en tiempo real'), defendió esta idea argumentando que los sistemas de telefonía no se habían cambiado en 100 años y que era una oportunidad única para crear un nuevo sistema de comunicación multimedia desde cero.

Después de varias modificaciones, la versión definitiva de SIP se publicaría en marzo de 1999 como la recomendación RFC2543. Debido al tremendo interés que despertó, el IETF creó nuevos grupos de trabajo, dando paso a la serie de recomendaciones desde la RFC3261 (mayo de 2002) a la RFC3266 (junio de 2002).

La gran ventaja que presenta SIP es que se trata de un protocolo pensado desde Internet para ser utilizado en Internet. Esto quiere decir que SIP aporta lo imprescindible para poder establecer una comunicación (multimedia o de datos) entre dos equipos de Internet, utilizando la misma filosofía que los protocolos de Internet más extendidos (HTTP, SMTP, etc.)

En las comparaciones con H.323, los defensores de SIP argumentan que se trata de una solución más flexible, simple, fácil de implementar, adaptable a dispositivos inteligentes y más apropiada para ofrecer características avanzadas. Quizás su mayor inconveniente es que no resulta tan compatible con los sistemas telefónicos tradicionales como lo es H.323. En cualquier caso, con el uso de *gateways*, las redes SIP pueden interconectarse con cualquier otro tipo de red.

Los programas Messenger de Microsoft o iChat de Apple utilizan el sistema SIP.

El resultado es que SIP está alcanzando unas cuotas de penetración en el mercado de telefonía IP cada vez mayores. De hecho, muchos consideran que SIP reemplazará completamente a H.323.

SIP puede establecer sesiones entre dos participantes, entre múltiples participantes (donde todos pueden hablar y escuchar) y en modo multidifusión (donde uno habla y los demás escuchan). De la misma forma, las sesiones pueden contener audio, vídeo o datos. Esta última opción permite que se pueda ejecutar cualquier tipo de aplicaciones simultáneamente a la comunicación de voz y vídeo (por ejemplo, para realizar presentaciones, compartir documentos o trabajar en grupo).

Otra curiosidad de SIP es que puede incluir campos opcionales con información específica del usuario. Esta posibilidad permite que los usuarios intercambien información no estándar, lo que hace posible el desarrollo de nuevos servicios. Por ejemplo, se podría incluir un texto en la invitación de la llamada que indicase el asunto: 'Sobre la reunión de mañana...'. En el caso de rechazar la llamada podría incluirse el motivo: 'Te llamo dentro de 10 minutos'. Por otro lado, si un usuario no está disponible, podría programar una contestación automática que dijera 'estaré disponible a partir de las 4 pm'.

LA ARQUITECTURA SIP

SIP es un protocolo de señalización que ha sido concebido para controlar el establecimiento, modificación y terminación de comunicaciones multimedia (audio, vídeo y datos). Por tanto, una vez establecida la comunicación, el intercambio de información multimedia se lleva a cabo mediante otros protocolos (generalmente RTP/RTCP). En cuanto al nivel de transporte, aunque SIP suele utilizar UDP, también funciona sobre TCP. Por otro lado, recientemente se ha presentado una propuesta para utilizar el protocolo SCTP (*Stream Control Transmission Protocol*, 'Protocolo de transporte de control de flujo', descrito en la recomendación RFC2960 de octubre de 2000).

Desde el punto de vista del nivel de red, SIP se pensó para funcionar sobre IP, no obstante, nada impide que pueda utilizarse sobre cualquier otro protocolo de red: ATM (*Asynchronous Transfer Mode*, Modo de transferencia asíncrono), FR (*Frame Relay*) o X25.

La arquitectura de SIP se basa en un modelo cliente-servidor en el que el cliente realiza solicitudes (*requests*) al servidor, quien le responde (*response*) para aceptar, rechazar o redirigir dichas solicitudes. Por ejemplo, en una comunicación directa entre terminales, el cliente sería el llamante y el servidor el llamado. El primero le solicita al segundo iniciar una sesión.

Además de los terminales de usuario, que pueden ser tanto clientes como servidores, la arquitectura SIP considera la existencia de otro tipo de servidores para poder crear redes multimedia más complejas.

La comunicación entre clientes y servidores se realiza mediante el envío de mensajes en modo texto, con una sintaxis muy similar a la de los mensajes HTTP. Esto quiere decir que los mensajes están compuestos por textos que son perfectamente legibles, lo que facilita las labores de desarrollo, así como la compatibilidad con las aplicaciones HTTP, que también se basan en textos legibles.

Componentes

Desde el punto de vista del equipamiento, la arquitectura SIP supone la existencia de dos tipos de equipos: terminales y servidores de red. Los terminales son los equipos finales que utilizan los usuarios, mientras que los servidores de red son equipos intermedios que aportan funcionalidad a las redes SIP.

Los equipos terminales disponen de dos componentes fundamentales:

- **Agente de usuario cliente** o UAC (*User Agent Client*). Se trata de la aplicación que permite que el terminal pueda realizar o iniciar una llamada. El UAC se utiliza para enviar solicitudes SIP.
- **Agente de usuario servidor** o UAS (*User Agent Server*). Se trata de la aplicación que permite que el terminal pueda recibir o responder a una llamada realizada por el UAC. El UAS recibe solicitudes SIP y devuelve las respuestas correspondientes en nombre del usuario. Estas respuestas pueden ser de aceptación, rechazo o redirección de la solicitud.

Los agentes de usuario le dan una funcionalidad completa al terminal, de forma que pueden utilizarse sin la intervención de los servidores de red. Por otro lado, podría haber terminales que contengan sólo uno de los agentes de usuario. En este caso, el terminal se utilizaría exclusivamente para escuchar o para transmitir.

En cuanto a los servidores de red, existen tres tipos:

- **Servidor *proxy*** (*proxy server*). Se trata de un equipo que recibe solicitudes del cliente, las analiza y decide el servidor al que debe reenviarlas. Si es necesario, el *proxy* puede modificar el mensaje de solicitud antes de reenviarlo. Para el siguiente servidor el mensaje procede del *proxy*. Una solicitud puede atravesar distintos servidores *proxy* antes de alcanzar su destino. En este caso, la respuesta atravesaría todo el camino inverso. Como el *proxy* maneja solicitudes y respuestas, necesita disponer, tanto de la funcionalidad cliente, como de la de servidor. Por otro lado, los servidores *proxy* pueden mantener un registro de todas las llamadas en curso o no (a estos modos se los conoce como *stateful* y *stateless* respectivamente). En el caso de que el *proxy* no mantenga un registro de la llamada se limitará a reenviar la solicitud y listo. El registro de llamadas es

necesario para poder ofrecer determinados servicios de valor añadido o, incluso, para determinadas aplicaciones de gestión.

- **Servidor de redireccionamiento** (*Redirect Server*). Estos servidores no reenvían los mensajes de solicitud, sino que le responde al cliente con la dirección, o direcciones, del servidor al que le tienen que enviar la solicitud. En este caso, es el cliente el que contacta directamente con el nuevo servidor. El servidor de redireccionamiento no contacta con otros servidores ni acepta llamadas (no tiene la funcionalidad de agente de usuario servidor).

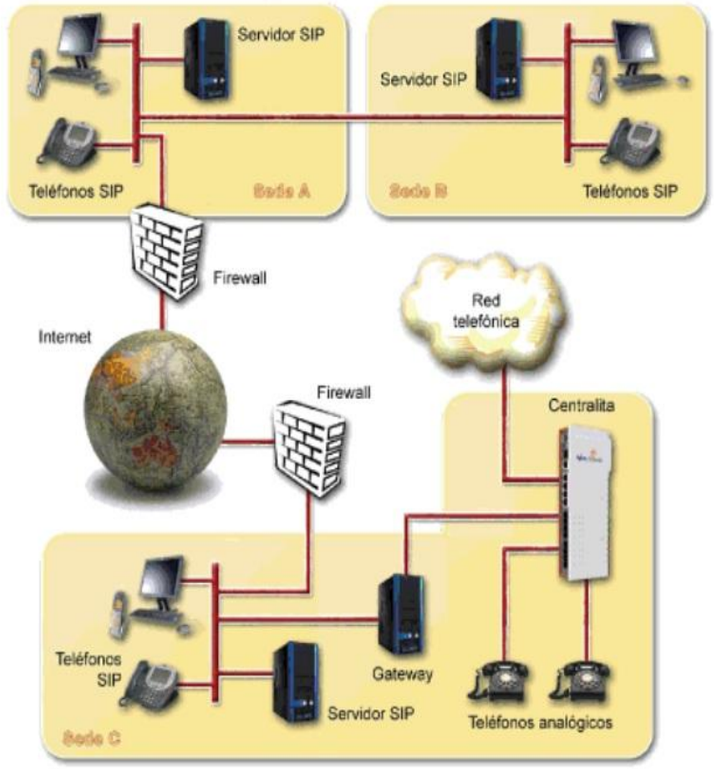

Figura 7.1. La arquitectura SIP

- **Servidor de registro** (*Registrar Server*). Este servidor mantiene un registro de la dirección SIP de un usuario y de su dirección IP correspondiente. Esto permite tener localizado a un usuario en todo momento. Por ejemplo, si un usuario dispone de distintos dispositivos SIP (ordenador de la oficina, PDA, ordenador de casa, etc.), cuando entra (*log-in*) en cualquiera de ellos, el

PROPIEDAD TÉCNICA PRIVADA

dispositivo le envía un mensaje al servidor de registro avisándole de su nueva dirección IP, con lo que el servidor ya sabe dónde tiene que reenviar las llamadas con destino a dicho usuario. A este servidor se lo conoce también como servidor de localización (*location server*)

Aunque los servidores de red podrían estar físicamente localizados en equipos distintos, suele ser normal que un mismo equipo tenga la funcionalidad de varios tipos de servidores, especialmente la combinación del servidor de registro con cualquiera de los anteriores.

URI. Las direcciones SIP

SIP permite la interconexión de equipos, tanto de una red IP como de otro tipo de redes, a través de *gateways*. Para que desde un equipo se pueda establecer una comunicación con otro, es necesario poder identificarlo de alguna forma. La red telefónica cuenta con un sistema de numeración internacional, Internet cuenta con el sistema de números IP y de nombres de dominio. El reto de SIP es establecer un sistema de identificación que sea compatible con distintas redes y que dé respuesta a todas las necesidades propias de un *gateway*. La solución es URI (*Uniform Resource Identifier*, 'Identificador uniforme de recursos'):

El formato de URI está definido en la recomendación RFC2396 y se basa en la estructura de direccionamiento empleada en Internet y conocida como URL (*Uniform Resource Locators*, 'Localizador universal de recursos'). De hecho, las direcciones URI tienen un formato similar al de una dirección de correo electrónico.

El formato general de una dirección URI es el siguiente:

sip:usuario@dominio;parámetro=valor

Este tipo de direccionamiento permite, por ejemplo, que se utilice un número telefónico tradicional en el campo usuario, identificándose en el campo dominio la dirección del *gateway* que interconecta con la red telefónica tradicional donde se encuentra dicho número. De la misma forma, el dominio de las direcciones URI podría ser también un número IP.

Por cierto, recuerde que el formato URL de una dirección de correo electrónico es *mailto:usuario@dominio*. Esto nos deja ver la similitud de ambos formatos.

URI SIP	EXPLICACIÓN
sip:usuario@dominio.tld	*Formato básico*
sip:usuario@dominio.com;transport=TCP	*Indicación del protocolo de transporte a utilizar (por defecto es UDP)*
sip:usuario@177.61.2.45	*Utilizando una dirección IP como dominio*
sip:+34-876543210@dominio.com;user=phone	*Utilizando un número telefónico como usuario*
sip:marketing@dominio.com;maddr=225.0.2.1;ttl=64	*Utilizando una dirección multicast. El parámetro ttl debe ser definido al utilizarse multicast y UDP*

Tabla 7.1. Ejemplo de formatos de direcciones URI

LOS MENSAJES SIP

Los mensajes del protocolo de señalización SIP son de dos tipos: solicitudes (*requests*) y respuestas (*responses*). Los clientes envían mensajes de solicitudes a los servidores; quienes responden a los clientes con mensajes de respuestas. El formato de los mensajes de solicitud y de respuesta es el mismo. Ambos disponen de una cabecera formada por distintos campos y de un cuerpo del mensaje.

Los mensajes se basan en el envío de textos utilizando la tabla de caracteres ISO 10646 y conocido como conjunto universal de caracteres (UCS, *Universal Character Set*). Esta tabla, al igual que la tabla ASCII, define el conjunto de ceros y unos que debe representar a cada carácter. No obstante, la ventaja de ISO 10646 frente a ASCII (más conocido en el mundo del PC) es que recoge todos los caracteres internacionales, incluidos los de caligrafías asiáticas.

Como los mensajes de SIP están formados por textos, esto quiere decir que si una persona viese el contenido de uno de estos mensajes, podría interpretarlo fácilmente (al igual que ocurre con el formato HTTP).

El formato general de los mensajes (de solicitud y respuesta) está definido en la recomendación RFC822, y se compone de:

- Una **línea de inicio** (*start-line*). Esta línea define el tipo de mensaje de que se trata. Existen dos tipos: línea de solicitud, utilizada por los mensajes de solicitud, y línea de respuesta, utilizada por los mensajes de respuesta.
- Una **cabecera** formada por uno o más campos de cabecera (*headers*). La cabecera se utiliza para incluir información adicional relativa a la solicitud o la respuesta. Por ejemplo, se incluye desde el remitente o el destinatario, hasta un texto indicando el motivo de la llamada.
- Una **línea en blanco** para indicar el final de la cabecera.

- El **cuerpo del mensaje** (*message-body*). SIP no define la estructura del cuerpo del mensaje ni se ocupa de su contenido. No obstante, normalmente, el cuerpo del mensaje se utiliza para describir el tipo de sesión que se va a establecer, versión del codec a utilizar, etc. Lo más habitual es utilizar la estructura definida por el protocolo SDP (*Session Description Protocol*, 'Protocolo de descripción de sesión'), aunque podría utilizarse cualquier otra. De hecho, el cuerpo del mensaje podría contener múltiples partes, cada una de ellas codificada con un protocolo distinto. Esta versatilidad hace posible utilizar SIP, por ejemplo, para transportar mensajes SS7 en formato binario, permitiendo, de esta forma, la interconexión con la red telefónica.

Figura 7.2. Ejemplo de mensaje SIP

La línea de solicitud

En el caso de los mensajes de solicitud, la línea de inicio recibe el nombre de línea de solicitud (*request-line*) y está formada por los siguientes componentes: una palabra que indica el tipo de solicitud de que se trata (método), dirección URI del servidor al que se dirige la solicitud e indicación de la versión del protocolo SIP utilizado. Estos tres componentes estarán separados por un espacio y terminarán con un carácter CRLF (un carácter particular de UCS que se conoce como *Carriage Return and Line Feed*, 'Retorno de carro y cambio de línea').

La sintaxis sería la siguiente:

*Método URI Versión-sip*CRLF

A los tipos de solicitudes se los conoce como métodos (*methods*) y están identificados por una palabra. En la versión 2.0 de SIP se incluyen seis tipos de solicitudes o métodos:

- ***INVITE*** (invitar). Este mensaje se utiliza para invitar a participar en una sesión (una llamada). En el cuerpo del mensaje se incluye el resto de parámetros necesarios: por ejemplo, el asunto de la llamada o el tipo de formato multimedia que puede utilizar el terminal llamante. Por su parte, el terminal llamado incluirá en el cuerpo del mensaje de su respuesta de aceptación los formatos multimedia que acepta.
- ***ACK*** (aceptación). Esta solicitud se utiliza para que el cliente (llamante) pueda confirmarle al servidor (llamado) que ha recibido su respuesta final de aceptación de participación en la sesión (llamada). Este método se utiliza exclusivamente con solicitudes INVITE, y no con otro tipo de solicitudes.
- ***OPTIONS*** (opciones). Se utiliza para preguntarle a un servidor por sus capacidades (los formatos multimedia que acepta, etc.). Se utiliza antes de iniciar la llamada para averiguar si admite determinadas características.
- ***BYE*** (adiós). Este mensaje se utiliza para terminar una sesión. Generalmente, se envía cuando cuelga el usuario.
- *CANCEL* (cancelar). Se utiliza para anular una solicitud para la que todavía no se ha recibido respuesta.
- ***REGISTER*** (registrar). Se utiliza para que un cliente pueda registrar su localización actual en un servidor SIP. El cliente puede registrarse en un servidor del que conoce su dirección o en todos los servidores SIP a través de la dirección *multicast* 224.0.1.175.

El siguiente podría ser un ejemplo de una línea de solicitud:

```
INVITE sip:coco@mumae.com SIP/2.0
```

Algunas extensiones de SIP definen métodos adicionales como: INFO, REFER o UPDATE. Por ejemplo, el método INFO, definido en RFC2976, se utiliza para transferir información durante el curso de una sesión (durante la llamada). Este método podría utilizarse, por ejemplo, para transmitir el coste de la llamada en curso, para retransmitir los tonos multifrecuencia marcados en el terminal o para informarle a un usuario prepago de que su saldo está a punto de acabarse.

La línea de respuesta

En el caso de tratarse de un mensaje de respuesta, éste informará del estado en el que se encuentra el servidor o si la solicitud se ha aceptado o rechazado. Las líneas de

inicio de los mensajes de respuesta se conocen como líneas de estado o línea de respuesta.

El tipo de respuesta se transmite con un código numérico conocido como código de estado (*status code*). La línea de estado está formada por tres componentes: versión del protocolo SIP, el código de estado y un texto con una breve explicación del código de estado incluido en el mensaje. Este texto ayuda a que una persona pueda interpretar directamente el mensaje de respuesta.

Al igual que ocurre con la línea de solicitud, los tres componentes de la línea de respuesta están separados por un espacio y termina con un carácter CRLF.

La sintaxis sería la siguiente:

*Versión-sip Código Texto*CRLF

Los códigos de estado tienen tres dígitos, donde el primero define la clase de respuesta y los otros dos el mensaje concreto dentro de la clase. Actualmente existen seis clases diferentes de mensajes de respuestas:

- **1xx**. De información (*informational*). Indica que se ha recibido la solicitud y que se está procesando. Por ejemplo, el mensaje 180 indica que está haciendo sonar el timbre del usuario.
- **2xx**. De aceptación (*Successful*). Indica que se ha recibido la solicitud y que todo está correcto para ejecutarla.
- **3xx**. Redirección (*Redirection*). Indica que el usuario llamado no está disponible en la dirección utilizada en la solicitud y que será redireccionada a la nueva dirección que se adjunta.
- **4xx**. Fallo en la solicitud (*Request Failure*). Indica que el mensaje de solicitud no es del todo correcto: existe algún error de sintaxis, no se está autorizado, etc.
- **5xx**. Fallo en el servidor (*Server Failure*). Indica que, aunque el mensaje de solicitud es válido, el servidor tiene algún problema para aceptarlo: servicio no disponible, saturación, error interno del servidor, etc.
- **6xx**. Fallo general (*Global Failure*). Indica que no se puede dar respuesta a la solicitud por algún tipo de fallo general.

Salvo las respuestas de información (1xx), el resto se consideran respuestas finales que dan por terminada la transacción SIP. Las respuestas 1xx son provisionales y no dan por terminada la transacción SIP.

El siguiente podría ser un ejemplo de una línea de estado:

```
SIP/2.0 200
```


CÓDIGO DE ESTADO	FRASE PROPUESTA (inglés)	FRASE PROPUESTA (español)
100	*Trying*	*En progreso*
180	*Ringing*	*Timbre sonando*
181	*Call is being forwarded*	*Llamada en desvío*
182	*Queued*	*En cola de espera*
183	*Session in progress*	*Sesión en progreso*
200	*OK*	*Todo bien*
300	*Multiple choices*	*Múltiples opciones*
301	*Moved permanently*	*Cambiado permanentemente*
302	*Moved temporarily*	*Cambiado temporalmente*
305	*Use proxy*	*Utiliza proxy*
380	*Alternative service*	*Servicio alternativo*
400	*Bad request*	*Solicitud incorrecta*
401	*Unauthorized*	*Sin autorización*
402	*Payment required*	*Se requiere un pago*
403	*Forbidden*	*Prohibido*
404	*Not found*	*No se encuentra*
405	*Method not allowed*	*Método no permitido*
406	*Not acceptable*	*No aceptable*
407	*Proxy authentication required*	*Se requiere autentificación del proxy*
408	*Request timeout*	*Expirado el tiempo de solicitud*
409	*Conflict*	*Conflicto*
410	*Gone*	*Recurso no disponible*
411	*Length required*	*Contenido insuficiente*
413	*Request entity too large*	*Solicitud mayor de lo esperado*
414	*Request-URI too long*	*Dirección URI demasiado larga*
415	*Unsupported media type*	*Tipo de medio no admitido*
416	*Unsupported URI scheme*	*Esquema URI no admitido*
420	*Bad extension*	*Extensión incorrecta*
421	*Extension required*	*Se requiere una extensión*
423	*Interval too brief*	*Intervalo demasiado breve*
480	*Temporarily not available*	*No disponible temporalmente*
481	*Call leg/transaction does not exist*	*La llamada no existe*
482	*Loop detected*	*Detectado un bucle*
483	*Too many hops*	*Demasiados saltos*
484	*Address incomplete*	*Dirección incompleta*
485	*Ambiguous*	*Ambiguo*
486	*Busy here*	*Ocupado*
487	*Request terminated*	*Solicitud terminada*
488	*Not acceptable here*	*Inaceptable*
491	*Request pending*	*Solicitud pendiente*
493	*Undecipherable*	*Indescifrable*
500	*Server internal error*	*Error interno del servidor*
501	*Not implemented*	*No implementado*
502	*Bad gateway*	*Gateway erróneo*
503	*Service unavailable*	*Servicio no disponible*
504	*Server timeout*	*Expirado el tiempo del servidor*
505	*SIP version not supported*	*Versión SIP no admitida*
513	*Message too large*	*Mensaje demasiado grande*
600	*Busy everywhere*	*Todo ocupado*
603	*Decline*	*Declinación*
604	*Does not exist anywhere*	*No existe*
606	*Not acceptable*	*No se acepta*

Tabla 7.2. Códigos de estado SIP

La cabecera

La cabecera de los mensajes está formada por campos, y tienen un formato similar al de las cabeceras de HTTP. Algunos de estos campos son utilizados en todos los tipos de mensajes, mientras que otros son específicos de un tipo de mensaje particular.

Por otro lado, existen campos de cabecera que pueden ser modificados o añadidos por los servidores *proxy* que intervienen en la comunicación, mientras que otros van dirigidos directamente a ser interpretados por el agente de usuario. A los primeros se los conoce como *hop-by-hop* (salto a salto) y a los segundos como *end-to-end* (extremo a extremo). Aunque el orden en el que aparecen los campos en la cabecera del mensaje es indiferente, los campos *hop-by-hop* deben aparecer antes que los campos *end-to-end*.

Inicialmente, la recomendación RFC2543 incluía 37 campos de cabecera, no obstante, en la actualidad ya hay definidos 46. Estos campos están divididos en cuatro grupos diferentes:

- **Campos generales** (*general headers*). Se utilizan, tanto en los mensajes de solicitudes como en los de respuesta, para incluir información básica necesaria para la gestión de los mensajes. Estos campos son: *Call-ID*, *Contact*, *Cseq*, *Date*, *Encryption*, *From*, *Organization*, *Retry-After*, *Subject*, *Supported*, *Timestamp*, *To*, *User Agent* y *Via*.
- **Campos de entidad** (*entity headers*). Definen información adicional relacionada con el cuerpo del mensaje (SDP) o, si no hubiera cuerpo del mensaje, sobre los recursos identificados en la solicitud. El propósito de estos campos es indicar el tipo y formato de la información incluida en el cuerpo del mensaje. Son estos: *Allow*, *Content-Encoding*, *Content-Length*, *Content-Type*, *Content-Disposition*, *Expires* y *MIME-Version*.
- **Campos de solicitud** (*request headers*). Los utilizan los clientes para incluir información adicional relacionada con la solicitud o con el propio cliente. Son los siguientes: *Accept*, *Accept-Encoding*, *Accept-Language*, *Accept-Contact*, *Authorization*, *Hide*, *In-Reply-To*, *Max-Forwards*, *Priority*, *Proxy-Authorization*, *Proxy-Require*, *Record-Route*, *Reject-Contact*, *Request-Disposition*, *Require*, *Response-Key*, *Route*, *Rack* y *Session-Expires*.
- **Campos de respuesta** (*response headers*). Los utilizan los servidores para incluir información adicional relacionada con la respuesta. Estos son: *Proxy-Authenticate*, *Server*, *Unsupported*, *Warning*, *WWW-Authenticate* y *Rseq*.

Si se está interesado en la definición de todos los campos, se puede recurrir a la recomendación RFC2543, no obstante, para hacernos una idea de su utilización será suficiente con ver el significado de algunos de ellos:

- ***Call-ID*** (Identificación de la llamada). Es un código que identifica cada llamada de forma unívoca, permitiendo, por ejemplo, hacer corresponder una respuesta con su solicitud, detectar duplicados de solicitudes *INVITE* o cambiar dinámicamente los parámetros de una llamada.
- ***Cseq*** (Identificación de la solicitud). Identifica cada solicitud. La respuesta a una solicitud incorpora el mismo Cseq.
- ***From*** (Desde). Aparece en todos los mensajes de solicitud y de respuesta para identificar el origen de la solicitud (terminal llamante).
- ***To*** (Para). Aparece en todos los mensajes de solicitud y de respuesta para identificar al destinatario de la solicitud (terminal llamado).
- ***Via***. Registra la ruta que sigue la solicitud. Por tanto, cada *proxy* por el que pasa la solicitud añade un campo *Via*.
- ***Content-Length*** (Longitud del contenido). Longitud del cuerpo del mensaje en octetos.
- ***Content-Type*** (Tipo de contenido). Indica el tipo de contenido incluido en el cuerpo del mensaje (normalmente será SDP, valor *application/sdp*).
- ***Content-Disposition*** (Disposición del contenido). Indica cómo debe interpretarse el cuerpo del mensaje. Algunos de los valores son: *session*, para indicar que se trata de una descripción de las características de una sesión (SDP), *icon*, para indicar que el cuerpo incluye una foto del llamante, *render*, para indicar que se trata de un texto para ser mostrado a la otra parte, o *alert*, para indicar que se trata de un audioclip o timbre personalizado.
- ***Subject*** (Asunto). Se trata de un texto que indica la naturaleza de la llamada.

El siguiente es un ejemplo de la cabecera de un mensaje:

```
INVITE sip:coco@mumse.com SIP/2.0
Via:SIP/2.0/UDP Cani.com
From: ana<sip:ana@delhi.mumse.com>
To: Marcos coco@mumse.com
Call-ID:1234567890@cani.com
Content-Type:application/sdp
Cseq:1 INVITE
Content-Length: 232
```

El cuerpo del mensaje. SDP

Los mensajes SIP no están obligados a incluir un cuerpo, sin embargo, cuando lo hacen, puede tratarse de información dirigida a cualquier aplicación del usuario o información de señalización. El contenido del cuerpo del mensaje no es analizado por los equipos intermedios de la red, sino que sólo es de interés para los equipos terminales (agentes de usuario).

CAMPOS	ACK	BYE	CAN	INV	OPT	REG
Accept	N	O	N	O	S*	O
Accept-Encoding	N	O	N	O	O	O
Accept-Language	N	O	N	O	O	O
Alert-Info	N	N	N	O	N	N
Allow	N	O	N	O	O	O
Authentication-Info	N	N	N	N	N	N
Authorization	O	O	O	O	O	O
Call-ID	S	S	S	S	S	S
Call-Info	N	N	N	O	O	O
Contact	O	N	N	O	O	O
Content-Disposition	O	O	N	O	O	O
Content-Encoding	O	O	N	O	O	O
Content-Languages	O	O	N	O	O	O
Content-Length	T	T	T	T	T	T
Content-Type	C	C	C	C	C	C
CSeq	S	S	S	S	S	S
Date	O	O	O	O	O	O
Error-Info	N	N	N	N	N	N
Expires	N	N	N	O	N	O
From	S	S	S	S	S	S
In-Reply-to	N	N	N	O	N	N
Max-Forwards	S	S	S	S	S	S
Min-Expires	N	N	N	N	N	N
MIME-Version	O	O	N	O	O	O
Organization	N	N	N	O	O	O
Priority	N	N	N	O	N	N
Proxy-Authenticate	N	N	N	N	N	N
Proxy-Authorization	O	O	N	O	O	O
Proxy-Require	N	O	N	O	O	O
Record-Route	O	O	O	O	O	O
Reply-to	N	N	N	O	N	N
Require	N	C	N	C	C	C
Retry-After	N	N	N	N	N	N
Route	C	C	C	C	C	C
Server	N	N	N	N	N	N
Subject	N	N	N	O	N	N
Supported	N	O	O	S*	O	O
Timestamp	O	O	O	O	O	O
To	S	S	S	S	S	S
Unsupported	N	N	N	N	N	N
User-Agent	O	O	O	O	O	O
Via	S	S	S	S	S	S
Warning	N	N	N	N	N	N
WWW-Authenticate	N	N	N	N	N	N

C = Condicional. Depende del contexto del mensaje N = No aplicable O = Opcional
S = Debe estar presente, pero el destinatario debe poder procesar la solicitud sin este campo*
S = Obligatorio T = Debe incluirse si se usa TCP

Tabla 7.3. Campos de cabecera a incluir en cada solicitud

CAMPOS	Código	ACK	BYE	CAN	INV	OPT	REG
Accept	2xx	N	N	N	O	S*	O
Accept	415	N	C	N	C	C	C
Accept-Encoding	2xx	N	N	N	C	S*	O
Accept-Encoding	415	N	C	N	C	C	C
Accept-Language	2xx	N	N	N	C	S*	O
Accept-Language	415	N	C	N	C	C	C
Alert-Info	180	N	N	N	O	N	N
Allow	2xx	N	O	N	S*	S*	O
Allow	405	N	S	N	S	S	S
Allow	Todos salvo 2xx, 405	N	C	N	O	O	O
Authentication-Info	2xx	N	O	N	O	O	O
Call-Info	Todos	N	N	N	O	O	O
Contact	1xx	N	N	N	O	N	N
Contact	2xx	N	N	N	S	O	O
Contact	3xx, 485	N	O	N	O	O	O
Content-Length	Todos	T	T	T	T	T	T
Content-Type	Todos	C	C	N	C	C	C
Error-Info	300-699	N	O	O	O	O	O
Min-Expires	423	N	N	N	N	N	S
Organization	Todos	N	N	N	O	O	O
Proxy-Authenticate	401	N	O	O	O	O	O
Proxy-Authenticate	407	N	S	N	S	S	S
Record-Route	2xx, 18x	N	O	O	O	O	N
Require	Todos	N	C	N	C	C	C
Retry-After	404, 413, 480, 486, 500, 503, 600, 603	N	O	O	O	O	O
Server	Todos	N	O	O	O	O	O
Supported	2xx	N	O	O	S*	S*	O
Unsupported	420	N	S	N	S	S	S
Warning	Todos	N	O	O	O	O	O
WWW-Authenticate	401	N	S	N	S	S	S
WWW-Authenticate	407	N	O	N	O	O	O

NOTAS: (1) Los campos a incluir dependen del código de estado y de la solicitud a la que se responde. (2) Los campos Call-ID, CSeq, From, To y Via son obligatorios para todos los códigos. (3) Los campos Content-Disposition, Content-Encoding, Content-Language, Date, MIME-Version, Timestamp y User-Agent son opcionales para todos los códigos.

C = Condicional. Depende del contexto del mensaje N = No aplicable O = Opcional
S = Debe estar presente, pero el destinatario debe poder procesar la solicitud sin este campo*
S = Obligatorio T = Debe incluirse si se usa TCP

Tabla 7.4. Campos de cabecera a incluir en cada respuesta

Como el cuerpo del mensaje puede utilizarse para distintas finalidades (algunas de ellas muy particulares), SIP no define su estructura, pudiéndose utilizar cualquier mecanismo que se estime oportuno. No obstante, cuando SIP utiliza el cuerpo del

mensaje para el intercambio de información de señalización, suele utilizar el protocolo SDP (*Session Description Protocol*, 'Protocolo de descripción de la sesión').

La información de señalización que incluye SIP en el cuerpo del mensaje se utiliza principalmente para acordar las características de la comunicación multimedia de acuerdo a las propiedades o preferencias configuradas en los equipos participantes. El cuerpo del mensaje viene a ser a SIP lo que el protocolo H.245 es a las comunicaciones H.323.

SDP no es un protocolo diseñado para SIP, sino que se puede utilizar en distintos entornos. El objeto de SDP es describir las características de una sesión multimedia, incluido el anuncio, invitación u otras formas de inicio de la sesión. No obstante, SIP sólo utiliza parte de las características de SDP. Por ejemplo, aunque SDP permite el intercambio de parámetros de reuniones periódicas con múltiples participantes y de duración limitada, esto no se utiliza con SIP, donde las comunicaciones suelen ser espontáneas y de duración ilimitada.

Aunque SDP se describió por primera vez en la RFC2327 (abril de 1998), desde entonces se han sugerido varias mejoras que están recogidas en distintas recomendaciones. Por otro lado, la recomendación RFC3264 describe cómo se utiliza SDP y SIP de forma conjunta.

SDP, al igual que SIP, intercambia información en modo texto utilizando la tabla de caracteres ISO10646 (codificación UTF-8, RFC2044).

La forma que tiene SIP de utilizar SDP consiste en que el terminal que origina la llamada incluye una descripción de todas sus características (dirección, puerto y formatos multimedia que acepta por orden de preferencia). El destinatario le responde con sus características y con una aceptación o rechazo de cada uno de los formatos propuestos. El resultado es un acuerdo entre las partes de los parámetros de la comunicación.

SDP distingue entre los parámetros que le afectan a la sesión y aquellos que le afectan a un medio particular (por ejemplo, audio, vídeo, datos, etc.). Entre los primeros se encuentra el nombre de la sesión o la identificación del remitente, mientras que entre los segundos se encuentra el tipo de medio, protocolo de transporte utilizado o el número de puerto.

SDP tiene definido todo un conjunto de campos que recogen todos los parámetros que se necesitan intercambiar. Un mensaje SDP se compone de líneas de texto donde cada línea representa un campo. La línea comienza por una letra, que identifica al campo, seguido del signo igual (=) y de su valor correspondiente. No debe haber ningún espacio en blanco ni antes ni después del signo igual. La sintaxis sería la siguiente:

campo=valor

Para definir algunos campos es necesario incluir más de un valor (subcampos). En estos casos, se incluirán los valores uno a continuación de otro, en un orden prefijado, y separados por un espacio.

Los campos que tienen que ver con la sesión se incluyen antes que los que tienen que ver con un medio. Por otro lado, existen campos que deben incluirse obligatoriamente (v, o, s, t, y m), mientras que otros son opcionales. Por último, algunos campos pueden utilizarse tanto a nivel de sesión como a nivel del medio.

SDP define que el orden en el que deben aparecer los campos es el siguiente:

- Para los campos de nivel de sesión: v, o, s, i, u, e, p, c, b, t, r, z, k y a.
- Para los campos de nivel del medio: m, i, c, b, k, a.

Por cierto, algunos campos de SDP no tienen sentido dentro de SIP (por ejemplo, s, i, u, t, r, z). En estos casos, o bien no se incluyen, o bien, de ser obligatorios, tomarán un valor sin aportar información.

Los campos de SDP son los siguientes:

- **v** (*version*). Versión del protocolo.
- **o** (*origin*, origen). Identifica al remitente y a la sesión. Tiene seis subcampos: nombre de usuario del remitente (suele tomarse el nombre de usuario configurado en el ordenador del remitente, aunque si no existe se incluye un guión -), identificador de sesión (identificador único creado por el ordenador del remitente), versión de la sesión, tipo de red (texto indicando el tipo de red, en el caso de Internet se utiliza IN), tipo de dirección (por ejemplo IP4 o IP6 para indicar la versión 4 o 6 del protocolo IP) y dirección (dirección de la máquina del remitente, puede indicarse como IP o como dominio).
- **s** (*session name*, nombre de la sesión). Se trata de un texto descriptivo que suele mostrarse a los participantes de la sesión. Este texto es especialmente interesante en las sesiones *multicast*. En el caso de SIP, donde las sesiones suelen ser *unicast* con sólo dos participantes, este texto no tiene sentido, por lo que suele utilizarse un espacio en blanco.
- **t** (*time*, hora). Facilita la hora predefinida de comienzo y terminación de la sesión. En las sesiones entre dos participantes con duración indefinida (SIP) el valor de este campo será t=0 0.
- **m** (*media type*, tipo de medio). Tiene cuatro subcampos: tipo de medio (audio, vídeo, aplicación, datos o control), puerto (número de puerto utilizado en destino), protocolo de transporte utilizado (RTP) y formatos multimedia aceptados (se incluye una lista de formatos multimedia por orden de preferencia). Los números de puerto dependen del tipo de conexión y del protocolo de transporte utilizado. Como en VoIP se utiliza el protocolo RTP sobre UDP, el número de puerto puede ser cualquier valor entre 1024 y 65535.

Por ejemplo, *m=audio 45678 RTP/AVP 15 3 0* indica que se desea recibir voz (audio) en el puerto 45.678 sobre RTP con codec del tipo 15 (G.728), 3 (GSM) o 0 (G.711 ley μ).

- i (*session information*, información de sesión). Se trata de un texto que describe el asunto de la sesión. Este campo ofrece más detalles que el nombre de la sesión. No obstante, como la cabecera de SIP ya incluye el campo asunto (*subject*), generalmente, no se utiliza este campo.
- u (URI) Se trata de una dirección URI donde obtener más información sobre la sesión. Esta información suele ser especialmente útil en multiconferencias acordadas en las agendas de los participantes. Como las sesiones SIP suelen establecerse sin agenda previa, este campo deja de tener utilidad en este caso.
- e (*email*). Dirección de correo electrónico de la persona responsable de la sesión.
- p (*phone*, teléfono). Teléfono de la persona responsable de la sesión.
- c (*connection*, conexión). Tiene tres subcampos: tipo de red, tipo de dirección y dirección de conexión. Tienen el mismo significado que los subcampos de o (origen) pero referido al destinatario.
- b (*bandwidth*, ancho de banda). Especifica el ancho de banda necesario en kilobits por segundo.
- r (*regularly*, periódico). Este campo se utiliza en las multiconferencias que se celebran de forma periódica para especificar cada cuanto tiempo y cuantas veces debe celebrarse esta conferencia. Este valor no tiene sentido con SIP.
- z (*timezone*, zona horaria). Indica la zona horaria utilizada. Este campo tiene sentido en las sesiones periódicas, por lo que no suele utilizarse con SIP.
- k (*encryption*, cifrado). Facilita una clave de cifrado o especifica el mecanismo por el que puede ser obtenida la clave para cifrar y descifrar la información multimedia.
- a (*additional attributes*, atributos adicionales). Se utiliza para describir atributos adicionales que sean de aplicación a la sesión o a un medio individual.

PONERLO TODO JUNTO

Bueno, ha llegado la hora de juntar todas las piezas anteriores para ver cómo se lleva a cabo el proceso de señalización para establecer una comunicación utilizando el protocolo SIP. Para empezar, pongamos el caso de una comunicación directa terminal a terminal, sin que intervengan servidores intermedios. En este caso, el proceso es bastante simple, y se compone de los siguientes pasos:

1. El terminal llamante le envía al llamado un mensaje de solicitud *INVITE* para invitarle a participar en una sesión (una llamada). En este mensaje se incluyen los parámetros propuestos para la comunicación.

2. Aunque existen distintas posibilidades de respuesta, dependiendo de las circunstancias, lo habitual es que el terminal llamado haga sonar el timbre para avisar a su usuario y le envíe al terminal llamante un mensaje de respuesta (código 180) para hacerle saber que está procediendo con la llamada.
3. Cuando el usuario llamado contesta, su terminal le envía al llamante un mensaje de respuesta *OK* (código 200).
4. Por último, el terminal llamante acepta la comunicación enviando un mensaje de solicitud *ACK*.
5. A continuación se pasa al intercambio de información multimedia (audio, video o datos) utilizando RTP/RTCP.
6. Cuando cuelga una de las partes su terminal envía un mensaje de solicitud *BYE* al otro extremo.
7. El terminal que recibe el mensaje *BYE* responde con mensaje *OK* (código 200) para confirmar que entiende que la comunicación ha concluido. En este punto, la llamada está terminada.

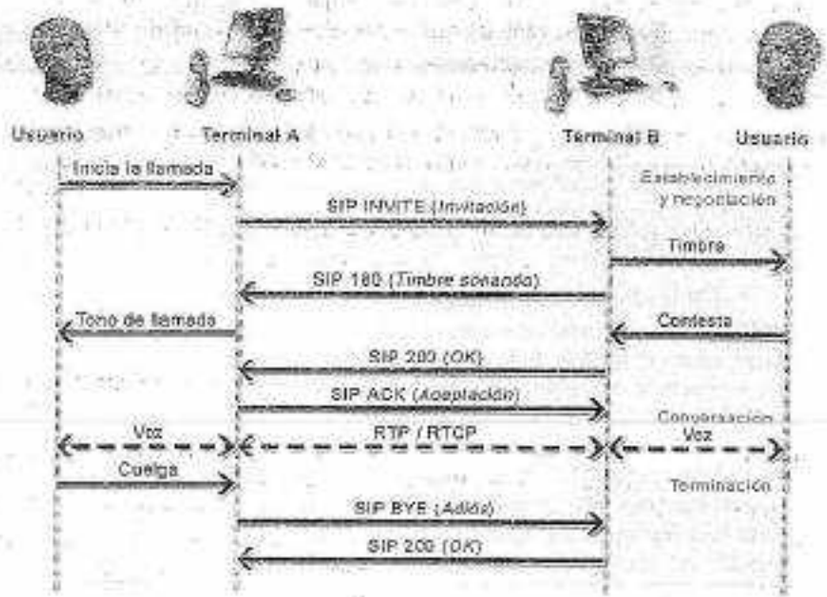

Figura 7.3. Desarrollo de una llamada con SIP

La invitación

Para escenificar este proceso con un ejemplo, supongamos que el usuario Juan (diaz@sistemas.talcual.com) quiere comunicarse con Luis (*gomez@finanzas.talcual.com*). El mensaje de invitación sería el siguiente:

```
INVITE sip:gomez@finanzas.talcual.com SIP/2.0
Via: SIP/2.0/UDP sistemas.talcual.com; branch=h3zG4aJ321
Max-Forwards: 70
From: Juan<sip:diaz@sistemas.talcual.com>;tag=44551
Contact: sip:diaz@sistemas.talcual.com
To: Luis<sip:gomez@finanzas.talcual.com>
Call-ID: 123456@sistemas.talcual.com
CSeq: 1 INVITE
Subject: Gastos
Content-Length: 218
Content-type: application/sdp
Content-Disposition: session

v=0
o=diaz 123456 001 IN IP4 sistemas.talcual.com
s=
c=IN IP4 sistemas.talcual.com
t=0 0
m=audio 4444 RTP/AVP 2
a=rtpmap 2 G726-32/8000
m=audio 4666 RTP/AVP 4
a=rtpmap 4 G723/8000
m=audio 4888 RTP/AVP 15
a=rtpmap 15 G728/8000
```

La primera línea es una línea de solicitud, a la que le siguen distintas líneas con los campos de cabecera. La línea en blanco separa la cabecera del cuerpo del mensaje, donde se relacionan los campos SDP que inician la negociación de los parámetros de la comunicación.

En este ejemplo, el equipo de Juan le está comunicando al de Luis que entiende los codecs G726-32, G723 y G728. A cada uno de ellos le asigna un número de puerto distinto (número de puerto propio al que el otro extremo tendría que dirigir la comunicación. En este ejemplo: 4444, 4666 y 4888 respectivamente), lo que le permitiría establecer más de una comunicación de audio simultánea. No obstante, si sólo se va a establecer una comunicación, podría especificarse el mismo puerto para todos ellos de la siguiente forma:

```
m=audio 4444 RTP/AVP 2 4 15
a=rtpmap 2 G726-32/8000
a=rtpmap 4 G723/8000
a=rtpmap 15 G728/8000
```


Negociar los parámetros y establecer la comunicación

En primer lugar, el terminal de Luis responde que está procediendo a avisar a su usuario con el siguiente mensaje:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP sistemas.talcual.com; branch=h3zG4aJ321
From:Juan<sip:diaz@sistemas.talcual.com>;tag=44551
To:Luis<sip:gomez@finanzas.talcual.com>;tag=11222
Contact: sip:gomez@finanzas.talcual.com
Call-ID :123456@sistemas.talcual.com
cSeq: 1 INVITE
Content-Length: 0
```

Una vez que Luis descuelga, su terminal enviará un mensaje de aceptación donde incluye los parámetros que acepta. En el ejemplo vamos a suponer que su terminal sólo entiende el codec G.728. Para cada codec no aceptado se especifica el número de puerto 0, mientras que para los codecs aceptados se facilita un número distinto de cero (6666 en este ejemplo). A partir de este momento, cada parte conoce el codec que se va a emplear y el número de puerto del terminal distante al que tiene que enviar la información multimedia:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP sistemas.talcual.com; branch=h3zG4aJ321
From:Juan<sip:diaz@sistemas.talcual.com>;tag=44551
To:Luis<sip:gomez@finanzas.talcual.com>;tag=11222
Contact:sip:gomez@finanzas.talcual.com
Call-ID:123456@sistemas.talcual.com
cSeq:1 INVITE
Subject:Gastos
Content-Length:163
Content-Type:application/sdp
Content-Disposition:session

v=0
o=gomez 45678 001 IN IP4 finanzas.talcual.com
s=
c=IN IP4 finanzas.talcual.com
t=0 0
m=audio 0 RTP/AVP 2
m=audio 0 RTP/AVP 4
m=audio 6666 RTP/AVP 15
a=rtpmap 15 G728/8000
```

Por último, el terminal de Juan acepta la comunicación dando paso al intercambio de información multimedia. El mensaje de aceptación sería el siguiente:

```
ACK sip:luis@finanzas.talcual.com SIP/2.0
Via: SIP/2.0/UDP sistemas.talcual.com; branch=h3zG4aJ321
Max-Forwards:70
From:Juan<sip:diaz@sistemas.talcual.com>;tag=44551
To: Luis<sip:gomez@finanzas.talcual.com>;tag=11222
Call-ID :123456@sistemas.talcual.com
cSeq: 1 ACK
Content-Length:0
```

Si el terminal de Luis aceptara más de un codec y no se decidiera por ninguno de ellos en particular, los incluiría en su respuesta asignándole un número de puerto distinto de cero. En este caso, el terminal de Juan decidiría el codec a utilizar generando un nuevo mensaje INVITE en el que incluirá sólo el codec elegido.

Si por el contrario, el terminal de Luis no pudiera aceptar ninguno de los codecs proporcionados por Juan, la respuesta incluiría un código de estado 488 (No aceptable) o 606 (No aceptable). Adicionalmente, esta respuesta podría incluir en su cabecera un campo de respuesta *Warning* con el código 304 (tipo de codec no disponible) o 305 (tipo de codec incompatible). En este caso, el llamante podría decidirse por enviar otro INVITE que incluya otros codecs distintos o informar a su usuario de que no es posible la comunicación.

Terminar la comunicación

Si Juan termina la llamada, su terminal enviará un mensaje BYE como el siguiente:

```
BYE sip:luis@finanzas.talcual.com SIP/2.0
Via:SIP/2.0/UDP sistemas.talcual.com; branch=h3zG4aJ321
Max-Forwards:70
From:Juan<sip:diaz@sistemas.talcual.com>;tag=44551
To:Luis<sip:gomez@finanzas.talcual.com>;tag=11222
Call-ID:123456@sistemas.talcual.com
cSeq:2 BYE
Content-Length:0
```

A lo que el terminal de Luis respondería como sigue:

```
SIP/2.0 200 OK
Via:SIP/2.0/UDP sistemas.talcual.com; branch=h3zG4aJ321
From:Juan<sip:diaz@sistemas.talcual.com>;tag=44551
To:Luis<sip:gomez@finanzas.talcual.com>;tag=11222
Contact:sip:gomez@finanzas.talcual.com
Call-ID:123456@sistemas.talcual.com
cSeq:2 BYE
Content-Length:0
```


Cuando intervienen servidores

Veamos ahora un ejemplo en el que intervienen servidores intermedios. Como en el ejemplo anterior, el usuario llamante, Juan, desea invitar a Luis a una comunicación. Al enviar esta invitación, el mensaje llega al servidor *proxy* del dominio de Luis. El *proxy* acepta este mensaje de invitación y contacta con el servidor de registro para conocer cuál es la localización exacta de Luis. El servidor de registro le informa al *proxy* de la dirección de Luis en este momento (la que previamente Luis había registrado). El *proxy* le envía al terminal de Luis una invitación, este terminal le avisa al usuario, quien descuelga. El terminal de Luis le envía una respuesta de aceptación al *proxy*, quien se lo retransmite al terminal de Juan. El terminal de Juan confirma la comunicación al *proxy* y éste se la retransmite al terminal de Luis, dando comienzo al intercambio de información multimedia.

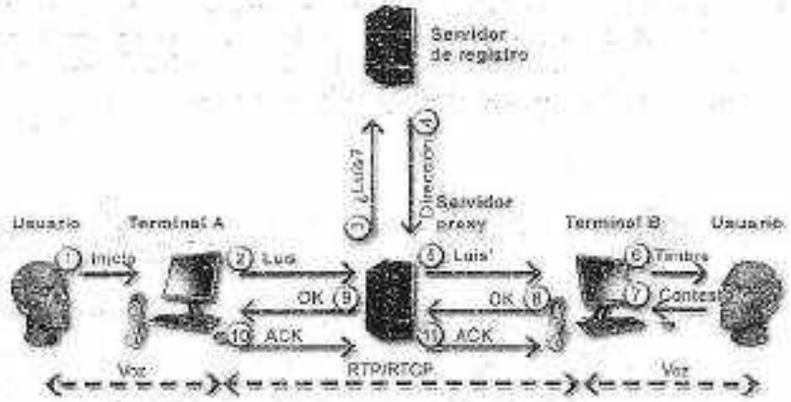

Figura 7.4. Proceso de llamada con servidor proxy y de registro

En el caso en el que, en lugar del *proxy*, se tuviera un servidor de redireccionamiento, este servidor averiguaría la dirección exacta de Luis y le devolvería al terminal de Juan una respuesta de redirección (clase 300) indicándole esta dirección. El terminal de Juan aceptaría este mensaje e iniciaría una comunicación directa con el terminal de Luis.

MEJORAS REALIZADAS A SIP

La recomendación que define SIP, RFC2543, se publicó en marzo de 1999. Desde entonces, se ha ido implementando en distintos entornos y desarrollando sus

aplicaciones. En muchas ocasiones se ha visto que, con pequeñas mejoras, se podría aumentar considerablemente el potencial de SIP. Todas estas mejoras están siendo estudiadas por el IETF, pero, hasta la fecha, no se ha publicado una nueva versión de SIP, aunque sí se dispone de recomendaciones independientes que recogen estas extensiones del protocolo.

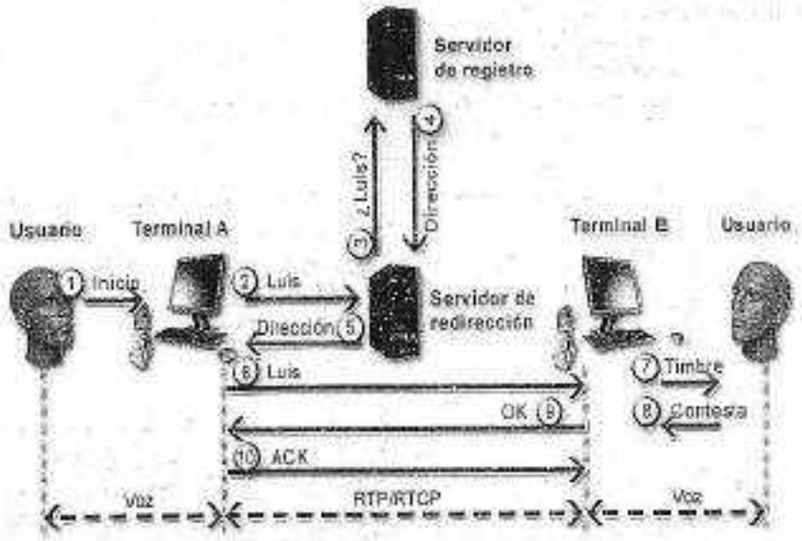

Figura 7.5. Proceso de llamada con servidor de redireccionamiento y de registro

Algunas de estas mejoras son las siguientes:

- **El método *INFO***. Este método se detalla en la RFC2976 y resuelve la necesidad del intercambio de información durante la comunicación. Por ejemplo, esto es útil para poder transmitir los tonos multifrecuencia de los dígitos telefónicos de extremo a extremo.
- **Notificación de eventos**. Algunas aplicaciones que utilizan SIP necesitan que sus usuarios puedan comunicar determinados eventos a la otra parte. Este es el caso, por ejemplo, de la mensajería instantánea o de un servicio de bolsa online. Esta necesidad se ha resuelto con la RFC3265. Esta recomendación añade dos nuevos métodos SIP: *SUBSCRIBE* (suscribir) y *NOTIFY* (notificar). El primero de ellos le permite a un usuario indicar los eventos de los que desea ser avisado y el segundo permite su notificación.

- **Mensajería instantánea**. El IETF ha establecido un grupo de trabajo (conocido como SIMPLE) especialmente dedicado a desarrollar las habilidades de SIP para su uso en aplicaciones de mensajería instantánea. La mensajería instantánea supone el intercambio de contenido en tiempo real entre los participantes. Aunque, generalmente, el contenido son mensajes cortos de texto, también se pueden intercambiar imágenes u otros datos binarios. Para dar respuesta a estas necesidades, SIMPLE ha propuesto añadir a SIP el método *MESSAGE* (mensaje). En este caso, el usuario A que quisiera enviar un mensaje instantáneo a un usuario B, enviaría un mensaje *MESSAGE* con el texto incluido en el cuerpo del mensaje. El terminal B respondería con un mensaje de aceptación (código 200). Si posteriormente el usuario B quisiera enviarle un texto de respuesta al A, le enviaría un nuevo mensaje *MESSAGE* independiente.
- **El método *REFER*** (referencia). Este método se utiliza para que un usuario pueda indicarle a otro que contacte con un tercero. La aplicación más evidente de este método es la transferencia de llamada. Por ejemplo, durante una comunicación entre Juan y Luis, este último se da cuenta de que con quien necesita hablar Juan es con Pedro, por lo que procede a transferirle la llamada, poniendo en comunicación a Juan con Pedro. En este caso, el terminal de Luis le envía un mensaje *REFER* al de Juan donde le incluye el campo de cabecera *Refer-to* (contactar con) con la dirección de Pedro. El terminal de Juan acepta el mensaje *REFER* de Luis con el código de respuesta 202 (aceptación, este código de estado no está definido en la versión original de SIP). Posteriormente, una vez establecida la comunicación con Pedro, le enviará un mensaje *NOTIFY* al terminal de Luis para indicarle que la transferencia se ha realizado con éxito. El terminal de Luis aceptará esta notificación con un código de respuesta 200. El método *REFER* también incluye el campo de cabecera *Referred-by* (referido por), que puede ser utilizado opcionalmente para informar sobre el usuario que realiza la transferencia.
- **Confiabilidad de las respuestas provisionales**. Existen mensajes provisionales de respuesta, como 180 (*ringing*) o 183 (*session progress*), que el terminal que los recibe no tiene que confirmar su recepción. Como estos mensajes se envían en UDP, podría ocurrir que nunca llegaran a su destino, produciendo que, bajo ciertas circunstancias, el establecimiento de la comunicación quedase en un punto muerto. Para resolver este problema, el IETF ha creado la recomendación RFC3262 donde se definen mensajes de confirmación de las respuestas provisionales. En este caso se incluyen dos nuevas cabeceras: RACK (*response ACK*, 'Aceptación de respuesta') y RSEQ (*Response sequence*, 'Secuencia de respuesta'). La primera es un campo a incluir en la cabecera de los mensajes de solicitud y la segunda en la cabecera de los mensajes de respuesta. También se define la *option tag 100rel* para ser utilizada en el campo *supported* de la cabecera. Por último, también se define el método PRACK (*Provisional Response ACK*, 'Aceptación de respuesta provisional').
- **El método *UPDATE***. Este método se ha pensado para poder cambiar los parámetros de una comunicación durante la misma. Para ello, durante el

transcurso de la sesión se envía un mensaje de solicitud *UPDATE* que contiene una oferta con los nuevos parámetros de sesión. La otra parte puede o no aceptar dichos parámetros.

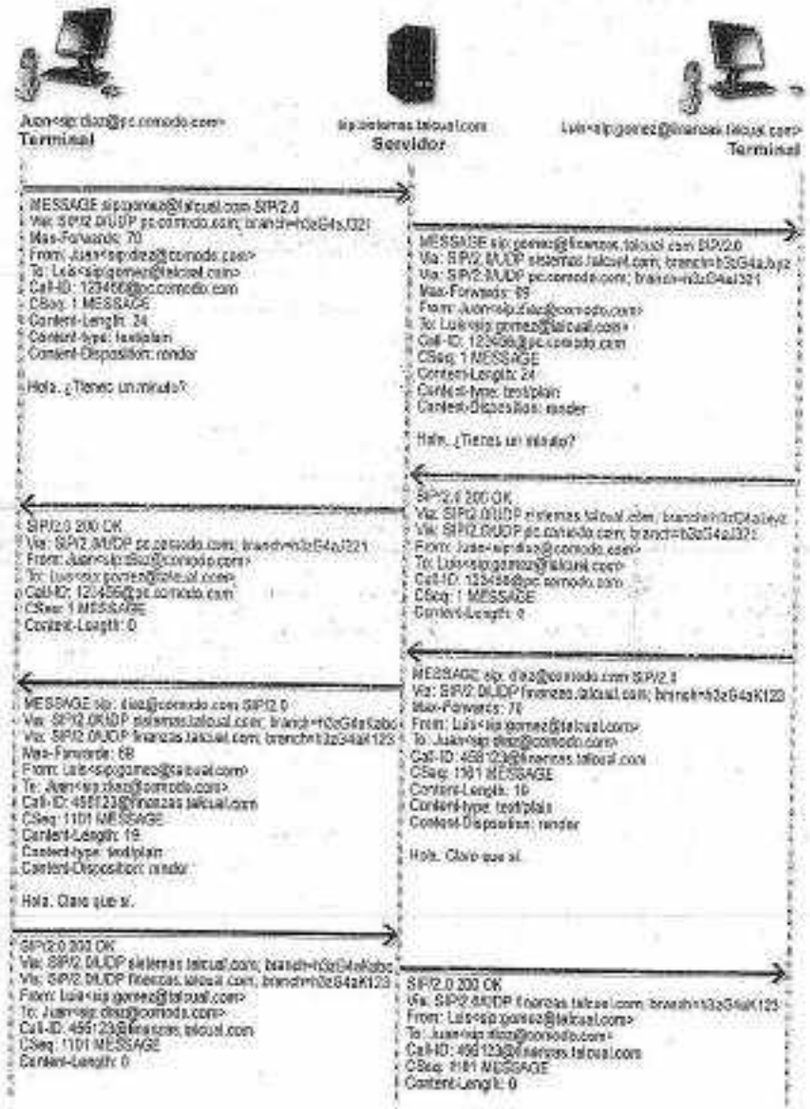

Figura 7.6. Ejemplo de mensajería instantánea con SIP

Interconexión con otras redes

INTRODUCCIÓN

Es evidente el gran valor que le aporta la tecnología IP al mundo de la telefonía, sobre todo el bajo coste y la integración de la voz con aplicaciones de datos. Esto nos lleva a pensar que las redes telefónicas evolucionarán hacia esta tecnología y que dentro de unos años habrán desaparecido las tradicionales redes de conmutación de circuitos. No obstante, lo cierto es que esta evolución se producirá lentamente. La red telefónica está ya desplegada y tiene unos niveles de calidad superiores a los de las actuales comunicaciones de VoIP. Por otro lado, los usuarios, acostumbrados a las comunicaciones de voz, no demandan masivamente la introducción de nuevos servicios, por lo que las operadoras no se ven en la necesidad de realizar una sustitución de redes a gran escala. Quiere esto decir que las redes de telefonía IP están condenadas a convivir con las de telefónica tradicional durante años.

Dado que ambas redes tienen que coexistir, lo lógico es que estén interconectadas de forma que los usuarios de los servicios de telefonía (tradicional o IP) no perciban, en lo posible, que realmente están interconectando dos tecnologías completamente diferentes. Esto quiere decir que los usuarios de una red deben poder comunicarse con los de la otra de forma completamente transparente, sin tener que hacer nada radicalmente distinto a lo que están acostumbrados. La interconexión de estas dos redes supone un doble reto:

- **Convertir la señal de la voz**. La red telefónica tradicional utiliza la codificación G.711 a 64 Kbps, mientras que los sistemas de VoIP utilizan distintos codecs con unas necesidades de ancho de banda menores.
- **Convertir la señalización**. La red telefónica tradicional utiliza el sistema SS7 (Sistema de Señalización número 7), mientras que las redes de VoIP utilizan protocolos como SIP o H.323.

La separación entre las señales de voz y las de control de la comunicación no es un concepto nuevo. Estos sistemas se desarrollaron en los años sesenta (Sistema de Señalización número 6 o SS6), aunque no se introducirían en la red telefónica de forma masiva hasta los años 80 con un estándar definido por la UIT-T, el Sistema de Señalización número 7 o SS7 (1981, serie Q.7xx). Anteriormente las señales de control se transmitían por el mismo canal de voz, mientras que SS7 se basa en disponer de canales específicamente dedicados a la señalización. Es lo que se conoce como sistemas de señalización por canal común o CCS (*Common Channel Signalling*). El sistema SS7 se utiliza en las redes telefónicas de todo el mundo (salvo excepciones que siguen utilizando algún sistema antiguo). Por su parte, las redes IP utilizan los sistemas de señalización SIP o H.323.

Si la interconexión con la red telefónica se realizara directamente con líneas telefónicas analógicas, la señal de la voz sería analógica y la señalización utilizada sería del tipo multifrecuencia.

Figura 8.1. Arquitectura de un gateway

EL *GATEWAY* Y EL MGC

Los equipos que permiten interconectar dos redes se conocen como *gateway* o pasarela. Un mismo equipo *gateway* puede llevar a cabo tanto la conversión de la señal multimedia como la de la señalización. No obstante, cuando el tráfico de interconexión es elevado se necesita utilizar más de un *gateway*. En este caso tiene sentido que las funciones de conversión de señal se sigan haciendo desde los *gateways*, conocidos como MG (*Media Gateway*), mientras que la lógica de la señalización se centraliza en un equipo independiente conocido como agentes de llamadas (*call agents*) o MGC (*Media Gateway Controller*).

La gran ventaja de esta separación es que desde un mismo equipo centralizado se pueden controlar múltiples *gateways* distribuidos en distintas localizaciones distantes. Esto permite, además, que modificar o introducir nuevas características sea tan simple como tocar un solo equipo (o muy pocos).

A los MGC también se los conoce como *Softswitch* (conmutador software). Este nombre deriva del hecho de que un software realiza la función que antiguamente realizaba un equipo electrónico de conmutación.

Los *gateways* se conectan por un lado con la red telefónica y por otro con la red IP. Para su interconexión con la red telefónica, se puede utilizar tanto líneas telefónicas individuales (analógicas o RDSI), como enlaces de varios canales (DS1, E1, RDSI 30B+D, etc.) o equipos especiales (por ejemplo, un sistema de locución o equipo de multiconferencia). En el lado de Internet se podría disponer de conexiones de banda ancha (DSL o cable) o enlaces de mayor caudal.

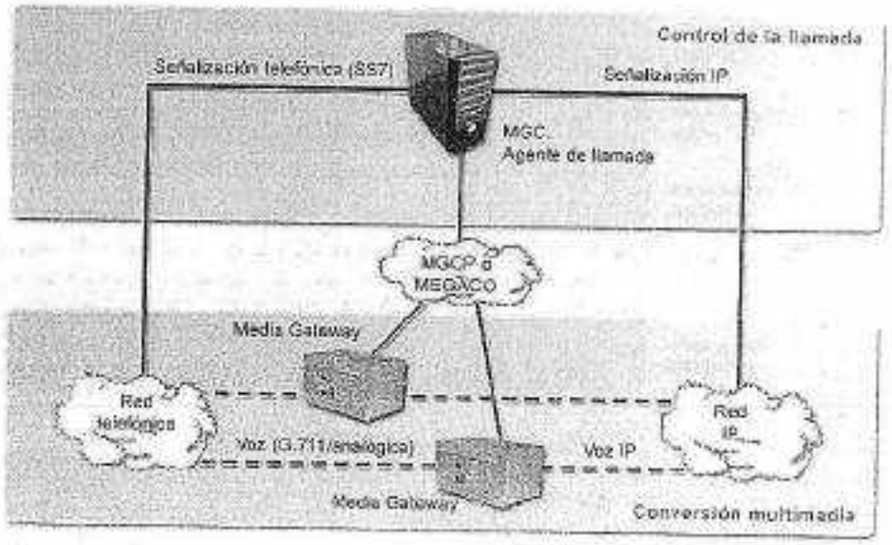

Figura 8.2. Interconexión con red telefónica

La interconexión entre los distintos MG, físicamente dispersos, y su agente de llamada o MGC, se realiza a través de IP utilizando un protocolo específico. Los más extendidos son:

- **MGCP** (*Media Gateway Control Protocol*), desarrollado por el IETF (RFC2705, octubre de 1999) y muy extendido hoy en día.
- **Megaco/H.248**, fruto de la colaboración entre el IETF (RFC3015 o Megaco) y la UIT-T (H.248). Se espera que este protocolo, surgido en junio de 2000, sea el dominante en un futuro.

Cuando un usuario de una red se intenta comunicar con alguien de la otra, la señalización le llega al MGC (agente de llamada), quien utiliza uno de estos protocolos para enviarle al *gateway* las órdenes adecuadas para el establecimiento de la comunicación. Los protocolos MGCP o Megaco también se utilizan para que el *gateway* mantenga informado al MGC del desarrollo de la comunicación, así como de cualquier otra circunstancia relacionada con la interconexión de ambas redes.

En el caso de que la estructura de la red requiera utilizar más de un MGC, la comunicación entre estos equipos se suele realizar mediante el protocolo SIP (aunque puede utilizarse otro).

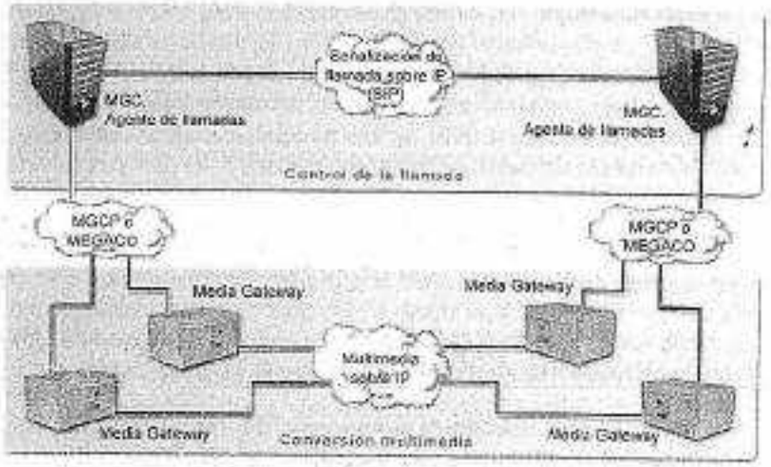

Figura 8.3. Interconexión entre MGC

MGCP

Los primeros protocolos creados para permitir la comunicación entre un MGC y un *gateway* fueron SGCP (*Simple Gateway Control Protocol*, 'Protocolo simple de control de *gateway*', febrero de 1998) y IPDC (*IP Device Control Protocol*, 'Protocolo de control de dispositivo IP', agosto de 1998). No obstante, estos protocolos se fusionaron en octubre de 1998 en uno nuevo conocido como MGCP. La versión definitiva de MGCP la publicaría el IETF en octubre de 1999 con la denominación RFC2705. Posteriormente la actualizaría en enero de 2003 (RFC3435).

MGCP es un protocolo maestro esclavo que permite que los MGC controlen las operaciones de los *gateways* mediante el intercambio de mensajes de órdenes y respuestas. El modelo MGCP contempla la existencia de dos componentes:

- **Puntos finales** (*endpoints*). Son fuentes de información multimedia, físicas o virtuales. Una fuente física es, por ejemplo, una línea telefónica analógica con un terminal telefónico, mientras que una fuente virtual sería un servicio contestador (software) residente en el mismo *gateway*. Cada punto final dispone de un identificador formado por el nombre de dominio del *gateway* al que está conectado y el nombre asignado al propio punto final. Este nombre puede estar formado por varias partes separadas por el carácter /. Por ejemplo, *E3/8/2@gateway.red.net*. MGCP admite utilizar comodines en la identificación de los puntos finales. Se utiliza $ para hacer referencia a un terminal cualquiera de entre los posibles y * para hacer referencia a todos ellos.

Entre los distintos tipos de puntos finales existentes se encuentran los siguientes: líneas telefónicas analógicas, líneas telefónicas RDSI, enlaces digitales DS1 o E1, servidor de locuciones, servicio de respuesta automática o IVR (*Interacting Voice Response*), puente de multiconferencia, enlace ATM, etc.

- **Conexiones**. Se trata de una asociación entre puntos finales sobre la que se transmiten los datos. Típicamente, una conexión será la relación establecida entre un punto final de la red telefónica tradicional y una sesión RTP/IP. Se pueden realizar conexiones punto a punto o punto-multipunto, pudiendo estar conectados los distintos puntos finales de la conexión a un mismo *gateway* o a *gateways* distintos.

Por tanto, desde el punto de vista del *gateway*, una llamada es una asociación entre una conexión y los puntos finales correspondientes, de forma que se pueda enviar y recibir información multimedia entre ellos.

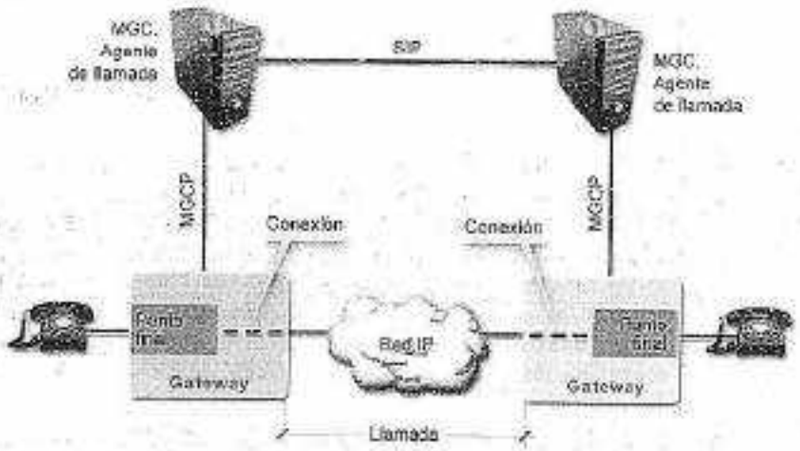

Figura 8.4. Interconexión de redes con MGCP

Para que el *gateway* se relacione adecuadamente con los puntos finales debe poder entender el comportamiento de los mismos. Por ejemplo, debe poder detectar que el terminal de la línea analógica ha descolgado, enviarle el tono de invitación a marcar y entender los dígitos que introduce su usuario. A cada uno de estos hechos se lo conoce con el nombre de **evento**. Son eventos: el envío o detección de cualquier tono en la línea o cualquier otra acción relacionada con el punto final. Los *gateways* pueden tanto detectar los eventos que ocurren en el punto final (detectar un descolgado o la marcación de un dígito) como generar acciones en los mismos (generar un tono de ocupado o el timbre de llamada).

Los eventos se agrupan en paquetes, de forma que el *gateway* implementa para cada tipo de punto final los paquetes que le correspondan. Por ejemplo, una línea telefónica analógica podría requerir la implementación de los paquetes relacionados con la detección y generación de tonos multifrecuencia.

MGCP identifica a cada paquete con una letra (por ejemplo, *L* para el paquete de línea analógica) y a cada evento por la identificación del paquete seguida de la del evento y separada por el carácter / (por ejemplo, *L/hd* para el evento descolgado). No obstante, el nombre del paquete se utiliza opcionalmente. Por otro lado, si se necesita identificar cualquier evento de un determinado paquete, se puede utilizar un carácter comodín (por ejemplo, *L/**). Esta solución resulta mucho más sencilla que listar todos y cada uno de los eventos posibles.

PAQUETE / Evento	SÍMBOLO
PAQUETE MULTIMEDIA GENÉRICO	*G*
Detectado un módem	*mt*
Detectado un tono de fax	*fx*
Tono de llamada	*rt*
Tono de congestión de red	*cg*
PAQUETE DTMF	*D*
Tono dtmf 0-9	*0-9*
*Tono dtmf #*ABCD*	*#*ABCD*
PAQUETE MF	*M*
Tono MF 0-9	*0-9*
PAQUETE ENLACE	*T*
PAQUETE LÍNEA	*L*
Transición a descolgado	*hd*
Transición a colgado	*hu*
Tono de invitación a marcar	*dl*
Tono de ocupado	*bz*
Timbre	*rg*
Tono de error	*e*
PAQUETE DE EMULACIÓN DE TERMINAL	*H*
PAQUETE RTP	*R*
PAQUETE DE SERVIDOR DE ACCESO A RED (NAS)	*N*
PAQUETE SERVIDOR DE AUDIO (CONTESTADOR)	*A*
PAQUETE SCRIPT	*Script*
Carga un script de Java	*java(url)*
Carga un script de XML	*xml(url)*
Completado	*oc*
Fallo	*of*

Tabla 8.1. Ejemplos de paquetes y eventos (lista no exhaustiva)

Las órdenes de MGCP

Los MGC controlan el comportamiento de los *gateways* mediante el envío de nueve tipos de órdenes (*commands*). Estas órdenes se envían en un mensaje con la

siguiente sintaxis: línea de orden, una o más líneas de parámetros y, opcionalmente, una descripción de la sesión (para esto se utiliza SDP).

Tanto la línea de orden como la de parámetros están separadas entre sí por el carácter CRLF (*Carriage Return and Line Feed*, 'Retorno de carro y cambio de línea'), mientras que la separación con la línea de descripción de la sesión es una línea en blanco.

La línea de orden está compuesta por un código de cuatro letras (ASCII) que identifican la orden, un identificador de la transacción (*TransactionID*), el punto final al que le afecta la orden (o del que procede) y la versión del protocolo (MGCP 1.0). Los elementos anteriores se separan entre ellos por un espacio en blanco:

Orden Transacción Punto_final MGCP 1.0

De las nueve órdenes, siete se utilizan para que el MGC le pase instrucciones al *gateway*, y sólo tres en sentido contrario: NTFY, RSIP y DLCX (que pueden ser utilizadas en ambas direcciones).

La lista de órdenes MGCP es la siguiente:

- **EPCF** (*Endpoint Configuration*, 'Configuración del punto final'). Se utiliza para que un MGC le informe al *gateway* de la codificación utilizada en la línea del punto final. Lo que indica realmente es si se trata de G.711 ley A o ley μ.
- **CRCX** (*Create Connection*, 'Crear conexión'). Se utiliza para indicarle al *gateway* que cree una conexión con un punto final.
- **MDCX** (*Modify Connection*, 'Modificar conexión'). Se utiliza para cambiar algunas de las características de una conexión existente.
- **DLCX** (*Delete Connection*, 'Eliminar conexión'). Se utiliza para indicarle al *gateway* que termine una conexión existente (o múltiples conexiones). Excepcionalmente, también se lo puede enviar el *gateway* al MGC para indicarle que la conexión se ha perdido.
- **RQNT** (*Notification Request*, 'Solicitud de notificación'). Utilizada por el MGC para pedirle al *gateway* que en el momento en que ocurra un determinado evento se lo notifique.
- **NTFY** (*Notify*, 'Notificación'). La envía un *gateway* para informarle al MGC de que ha ocurrido un evento determinado.
- **AUEP** (*Audit Endpoint*, 'Auditoría de punto final'). Utilizada por el MGC para preguntarle al *gateway* sobre el estado de un punto final o sobre cualquier otra circunstancia relacionada con el punto final. El *gateway* responderá con la información solicitada.
- **AUCX** (*Audit Connection*, 'Auditoría de conexión'). Utilizada por el MGC para preguntarle al *gateway* sobre una conexión específica.

- **RSIP** (*Restart In Progress*, 'Realizándose un reinicio'). Se utiliza para que el *gateway* le informe al MGC de que uno o más puntos finales se están conectando o desconectando.

Los parámetros

En cuanto a los parámetros, éstos se identifican por un código formado por uno o dos caracteres en mayúsculas. Al código le sigue el carácter dos puntos (:), un espacio y el valor del parámetro. Este valor puede ser un número, una cifra hexadecimal o un conjunto de valores separados por una coma. Cada orden tiene definido los parámetros que debe incluir de forma obligatoria, los que puede incluir opcionalmente y aquellos que no puede incluir.

Los principales parámetros definidos por MGCP son los siguientes (el resto pueden consultarse en RFC3435):

- **B** (*Bearer Information*, 'Información del portador'). Indica si la codificación utilizada en la línea es A o μ (mu). Por ejemplo, *B: e:mu*.
- **C** (*Call Id*, 'Identificación de la llamada'). Se trata del identificador único de la llamada.
- **A** (*Capabilities*, 'Características'). Se envía como respuesta a una pregunta sobre las características admitidas por el punto final. Cada característica se identifica por una o dos letras seguidas de dos puntos y del valor de la misma. Se puede indicar más de una característica separándola por una coma. Por ejemplo, *A: e:off, s:on* indica que no tiene cancelación de eco y que tiene activada la supresión de silencio.
- **I** (*Connection ID*, 'Identificación de la conexión'). Identificador único de la conexión de un punto final. Se trata de una cifra hexadecimal.
- **M** (*Connection Mode*, 'Modo de la conexión'). Define el modo de operación de la conexión. Algunos de estos modos son: inactivo (*inactive*), sólo enviar (*sendonly*), sólo recibir (*recvonly*), enviar y recibir (*sendrecv*) o conferencia (*confrnce*).
- **P** (*Connection Parameters*, 'Parámetros de la conexión'). Se trata de información estadística relacionada con la conexión. Esta información se le envía al MGC al final de la llamada como respuesta a un comando DLCX o AUEP.
- **T** (*Detect Events*, 'Eventos detectados'). Lista de eventos a detectar en un punto final.

OPCIÓN	SIGNIFICADO	DESCRIPCIÓN
PS	*Packets sent*	*Número de paquetes enviados.*
OS	*Octects sent*	*Número de octetos enviados.*
PR	*Packets received*	*Número de paquetes recibidos.*
OR	*Octecs received*	*Número de octetos recibidos.*
PL	*Packets lost*	*Número de paquetes no recibidos (se deducen por los saltos en el número de secuencia).*
JI	*Jitter*	*Fluctuación media del retardo en milisegundos.*
LA	*Latency*	*Latencia media en milisegundos.*

Tabla 8.2. Valores incluidos en el parámetro P

- **D** (*Digit Map*, 'Mapa de dígitos'). Es una representación de un plan de numeración donde se especifican las combinaciones de dígitos que pueden marcarse. El objetivo es que el *gateway* conozca las numeraciones que puede recibir del llamante. El mapa de dígitos lo envía el MGC dentro de una orden RQNT.
- **ES** (*Event States*, 'Estados de los eventos'). Se trata de una lista de eventos separados por comas. Se envían como respuesta a las órdenes de auditoría.
- **LC** (*Local Connection Descriptor*, 'Descriptor de la conexión local'). Se trata de una descripción SDP de la sesión con el punto final. Se indica la dirección IP y número de puerto desde el que el punto final espera recibir información multimedia, así como su formato.
- **L** (*Local Connection Options*, 'Opciones de la conexión local'). Se trata de una lista de opciones de la conexión (como el ancho de banda, periodo de paquetización, ganancia, cifrado, etc). Estas características se las sugiere el MGC al *gateway* cuando se va a establecer una conexión. Cada parámetro se identifica por uno o dos caracteres, seguido de dos puntos y su valor correspondiente. Por ejemplo, *L: e:off, s:on*.
- **MD** (*Max MGCP Datagram*, 'Datagrama MGCP máximo'). Indica el tamaño máximo de los paquetes MGCP que admite el MG. Este parámetro se puede incluir en la respuesta a un comando AUEP.
- **N** (*Notified Entity*, 'Entidad notificada'). Dirección del MGC.
- **O** (*Observed Events*, 'Eventos observados'). Lista de eventos detectados en un punto final. Se utiliza para informar al MGC de los eventos detectados.
- **PL** (*Package List*, 'Lista de paquetes'). Lista de paquetes de eventos admitidos por el punto final.
- **Q** (*Quarantine Handling*, 'Manejo de cuarentena'). Indican como debe manejar un punto final los eventos que ocurren mientras que el *gateway* está esperando una respuesta a una orden de notificación que le ha enviado a un MGC (a este tiempo se le llama periodo de cuarentena). Las opciones son que el punto final puede responder al evento o ignorarlo.

OPCIÓN	SIGNIFICADO	DESCRIPCIÓN
p	*Packetization Period*	*Periodo de paquetización en milisegundos. Si se especifica un rango estará formado por dos números separados por un guión.*
a	*Compression Algorithm*	*Tipo de codec utilizado. Por ejemplo a: G726.*
b	*Bandwidth*	*Ancho de banda en kbps.*
e	*Echo Cancellation*	*Cancelación de eco on/off.*
gc	*Gain Control*	*Ganancia en decibelios. También puede tomar el valor 'auto'*
s	*Silence Suppression*	*Supresión de silencio on/off*
t	*Type of Service*	*Tipo de servicio. Se indica con un código formado por dos dígitos hexadecimales.*
r	*Resource Reservation*	*Reserva de recursos. Puede tomar los valores g (servicio garantizado), cl (carga controlada) o be (best effort o hacer lo posible).*
k	*Encryption Key*	*Clave de cifrado*
nt	*Type of network*	*Tipo de red (IN, ATM o LOCAL).*

Tabla 8.3. Valores que pueden ser incluidos en los parámetros L y A

- **E** (*Reason Code*, 'Código de motivo'). Lo utiliza el *gateway* para indicar el motivo de la eliminación de una conexión o de un reinicio.
- **RC** (*Remote Connection Descriptor*, 'Descriptor de la conexión remota'). Descripción SDP del otro extremo. Este parámetro indica la dirección IP y número de puerto al que se enviará la información multimedia, así como el formato multimedia utilizado.
- **R** (*Requested Events*, 'Eventos solicitados'). Lista de eventos a controlar. Junto con estos eventos se informa de la acciones a realizar (notificar al MGC, emitir tono de llamada, etc.).
- **F** (*Requested Info*, 'Información solicitada'). Información facilitada por el *gateway* al MGC como respuesta a una orden de auditoría.
- **X** (*Request Identifier*, 'Identificador de solicitud'). Utilizado en las órdenes RQNT y NTFY para relacionar una notificación del *gateway* con la solicitud de información correspondiente.
- **K** (*Response ACK*, 'Confirmación de respuesta'). Incluye una lista de una o más identificaciones de transacción.
- **RD** (*Restar Delay*, 'Retardo de reinicio'). Indica el tiempo en segundos que falta para que un punto final vuelva a estar en servicio.
- **RM** (*Restart Method*, 'Método de reinicio'). Indica si el reinicio de un punto final es forzado o no, así como si el punto final va a volver a estar en servicio.
- **I2** (*Second Connection Id*, 'Identificación de la segunda conexión'). Identifica a la segunda conexión en los casos en los que se realiza una conexión entre dos puntos finales de un mismo *gateway*.
- **Z2** (*Second Endpoint Id*, 'Identificación del segundo punto final'). Identifica al segundo punto final en los casos en los que se realiza una conexión entre dos puntos finales de un mismo *gateway*.

PARÁMETROS	EPCF		CRCX		MDCX		DLCX		RQNT		NTFY		AUEP		AUCX		RSIP	
	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta	Comando	Respuesta
B	O[2]	N	O	N	O	N	O	N	O	N	N	N	N	O	N	N	N	N
C	N	N	S	N	S	N	O	N	N	N	N	N	N	N	N	O	N	N
A	N	N	N	N	N	N	N	N	N	N	N	N	N	O	N	N	N	N
I	N	N	N	O[2]	S	N	O	N	N	N	N	N	N	O	S	N	N	N
M	N	N	S	N	O	N	N	N	N	N	N	N	N	N	N	O	N	N
P	N	N	N	N	N	N	O[1]	O[1]	N	N	N	N	N	N	N	O	N	N
T	N	N	O	N	O	N	O	N	O	N	N	N	N	O	N	N	N	N
D	N	N	O	N	O	N	O	N	O	N	N	N	N	O	N	N	N	N
ES	N	N	N	N	N	N	N	N	N	N	N	N	N	O	N	N	N	N
L	N	N	O	N	O	N	N	N	N	N	N	N	N	N	N	O	N	N
MD	N	N	N	N	N	N	N	N	N	N	N	N	N	O	N	N	N	N
N	N	N	O	N	O	N	O	N	O	N	O	N	N	O	N	O	N	O
O	N	N	N	N	N	N	N	N	N	N	S	N	N	O	N	N	N	N
PL	N	O[1]	N	O[2]	N	O[1]	N	O[1]	N	O[1]	N	O[1]	N	O	N	O[1]	N	O[1]
Q	N	N	O	N	O	N	O	N	O	N	N	N	N	O	N	N	N	N
E	N	N	N	N	N	N	O	N	N	N	N	N	N	O	N	N	O	N
R	N	N	O	N	O	N	O	N	O[3]	N	K	N	N	O	N	N	N	N
X	N	N	O[4]	N	O[4]	N	O[4]	N	S	N	S	N	N	O	N	N	N	N
F	N	N	N	N	N	N	N	N	N	N	K	N	O	N	S	N	N	N
K	O	O[1]	O	O[1]	O	O[1]	O	O[1]	O	O[1]	O	O[1]	O	O[1]	O	O[1]	O	O[1]
RD	N	N	N	N	N	N	N	N	N	N	N	N	N	O	N	N	O	N
RM	N	N	N	N	N	N	N	N	N	N	N	N	N	O	N	N	S	N
I2	N	N	N	O	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Z2	N	N	O	O	N	N	N	N	N	N	N	N	N	N	N	N	N	N
S	N	N	O	N	O	N	O	N	O	N	N	N	N	O	N	N	N	N
Z	N	N	N	O	N	N	N	N	N	N	N	N	N	O[1]	N	N	N	N
RC	N	N	O	O[4]	O	O	N	N	N	N	N	N	N	N	N	O	N	N
LC	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	O	N	N

S = Parámetro obligatorio con esta orden
O = Parámetro opcional para esta orden
N = Parámetro prohibido con esta orden
2 = Pueden ser obligatorios o prohibidos bajo ciertas circunstancias
1 = Sólo puede incluirse bajo ciertas circunstancias
3 = Si no se incluye, la lista se considerará vacía
4 = El parámetro será obligatorio bajo ciertas circunstancias

Tabla 8.4. Parámetros incluidos en cada comando

- S (*Signal Requests*, 'Solicitudes de señal'). Identifica la señal (tono de invitación a marcar, llamada, etc.) que el *gateway* debe generar en el punto final.
- Z (*Specific EndpointId*, 'Identificación de punto final concreto'). Indica el punto final utilizado. Se suele utilizar cuando el MGC utiliza un carácter comodín en la identificación del punto final. En este caso, el *gateway* elige el punto final a utilizar y se lo comunica al MGC incluyendo este parámetro en un mensaje de respuesta.

CÓDIGO	DESCRIPCIÓN
000	*Confirmación de respuesta.*
100	*La transacción se está llevando a cabo. Se enviará un mensaje cuando termine la transacción.*
101	*La transacción está en cola. Se enviará un mensaje cuando termine la transacción.*
200	*Se ha completado la transacción solicitada.*
250	*La conexión ha sido eliminada.*
400	*La transacción podría no realizarse debido a un error temporal.*
401	*El terminal telefónico está ya colgado.*
402	*El terminal telefónico está ya descolgado.*
403	*La transacción no se ha ejecutado debido a que el punto final no dispone de suficientes recursos en este momento.*
404	*Ancho de banda insuficiente en este momento.*
405	*La transacción no se ha ejecutado debido a que el punto final se está reinicializando.*
406	*Ha expirado el tiempo. La transacción no se ha completado en un tiempo razonable y se ha abortado.*
407	*Transacción abortada debido a alguna razón externa.*
409	*No se ha ejecutado la transacción debido a una sobrecarga interna.*
410	*No hay un punto final disponible (se utilizó el comodín $).*
500	*No se ejecutó la transacción porque no se conoce el punto final.*
501	*No se ejecutó la transacción porque el punto final no está listo.*
502	*No se ejecutó la transacción porque el punto final no tiene suficientes recursos.*
503	*Comodín (*) demasiado complejo.*
504	*Comando desconocido o no soportado.*
510	*No se ha ejecutado la transacción debido a un error en el protocolo.*
511	*No se ha ejecutado la transacción debido a que el comando incluye una extensión desconocida.*
512	*No se ha ejecutado la transacción debido a que el gateway no está equipado para detectar el evento solicitado.*
515	*La transacción hace referencia a un ConnectionID incorrecto.*
516	*La transacción hace referencia a un CallID incorrecto.*
521	*Punto final redirigido a otro agente de llamada.*
528	*Versión del protocolo incompatible.*
534	*Fallo en la negociación del codec.*
535	*Periodo de paquetización no admitido.*
539	*Parámetros de la orden inválido o no admitido.*

Tabla 8.5. Códigos de respuestas MGCP (lista incompleta)

Las respuestas a las órdenes

A cada mensaje MGCP que contiene una orden le sigue su mensaje de respuesta correspondiente. El mensaje de respuesta se compone de una línea de respuesta seguida, opcionalmente, por uno o más parámetros de respuesta. La línea de respuesta está formada por un código seguido del identificador de transacción (*TransactionID*) y, opcionalmente, seguido por un comentario. Cada una de estas partes está separada por un carácter de espacio. La línea de respuesta se termina con un carácter CRLF:

Código Transacción ComentarioCRLF

El código de respuesta es un valor numérico entero perteneciente a una de las siguientes categorías:

- 0XX. Confirmación de respuesta.
- 1XX. Respuesta provisional. La respuesta definitiva vendrá posteriormente.
- 2XX. La orden se ha completado con éxito.
- 4XX. Fallo debido a un error transitorio.
- 5XX. Fallo debido a un error permanente.
- 8XX. Respuesta específica.

El mensaje de respuesta debe seguir al de la orden que lo originó y utilizará la misma identificación de transacción (*TransactionID*). Las respuestas pueden incluir los mismos parámetros que las órdenes, aunque la lista de parámetros permitidos, opcionales y prohibidos es diferente.

Iniciar una llamada con un MGC

Una vez descrito el modelo de funcionamiento del protocolo MGCP podemos recrear el proceso de inicio de una llamada telefónica entre dos redes. Hay que tener en cuenta que el MGC puede estar conectado directamente con el sistema de señalización por canal común (SS7) de la red telefónica tradicional o, si no fuese así, puede recibir las informaciones de señalización a través del *gateway*. Un *gateway* conectado a una línea telefónica analógica está configurado para que detecte que el punto final ha descolgado (evento *hd* del paquete *L*), envía un tono de invitación a marcar (evento *dl*), recibe los dígitos marcados (eventos del *0* al *9*), etc. Estas informaciones se las notifica al MGC para que inicie la llamada después de tomar las decisiones oportunas de enrutamiento.

En este ejemplo vamos a suponer que tenemos dos puntos finales (dos terminales) conectados a dos *gateways* diferentes que pertenecen a un mismo MGC. Los pasos a dar son los siguientes:

1. El MGC recibe información de la llamada telefónica. Esta información puede recibirla a través del *gateway* del punto final llamante o del sistema de señalización de la red origen de la llamada.

2. Identificados los puntos finales y *gateways* correspondientes, el MGC utiliza el comando CRCX (crear conexión) para indicarle al *gateway* A que cree una conexión con el punto final 1 (*PF1@gwA.redvoip.net*). Como se ha visto anteriormente, este comando debe incluir la identificación de la transacción (*1045*) y de la llamada (*C: 9876543*). También detalla la codificación utilizada (*B: e:mu*) y que el modo de la conexión es de recibir solamente (*M: recvonly*). Se fija temporalmente en este modo porque todavía no se dispone de la identificación del otro extremo (dirección IP y puerto donde enviar la información multimedia). Ya se modificará posteriormente el modo de conexión cuando se conozcan los detalles de la sesión.

Figura 8.5. Iniciar una llamada con un MGC

3. El *gateway* A envía una respuesta CRCX con la misma identificación de transacción, una identificación de conexión (I: AAAA) y una descripción de la sesión indicando el codec a utilizar (en este caso G.711 ley mu, indicado por el código 0 de RTP/AVP).

4. El MGC envía un comando CRCX al *gateway* B indicándole el punto final (*PF2@gwB.redvoip.net*), la identificación de la transacción (*2022*), y la misma identificación de llamada (*C: 9876543*). También incluye el parámetro RC con la descripción SDP (número IP, puerto, etc.) del punto final 1 que recibió en el mensaje de respuesta del *gateway* A. En este caso, el modo de conexión indicado es enviar y recibir (*M: sendrecv*) ya que existe una descripción del otro extremo.

5. El *gateway* B responde positivamente incluyendo la identificación de la conexión (*I: BBBB*) y los detalles SDP de su punto final (parámetro LC).

6. El MGC le envía una orden MDCX (modificar conexión) al *gateway* A para indicarle los datos finales de la conexión. El comando incluye especialmente el parámetro RC con los datos SDP del otro extremo y el parámetro M indicándole que cambie el modo de conexión a enviar y recibir (*sendrecv*).

7. El *gateway* A le envía al MGC una respuesta positiva a este comando MDCX.

8. A partir de este momento puede comenzar el intercambio de información multimedia.

Iniciar una llamada con dos MGC

En el caso de que los *gateways* implicados en la llamada pertenecieran a dos MGC distintos, esto requeriría un intercambio de información entre ambos MGC. Este intercambio se realiza normalmente utilizando el protocolo SIP. La gran ventaja de SIP frente a otras opciones es que tanto MGCP como SIP utilizan SDP para describir la sesión, facilitando, de esta forma, el entendimiento.

En este modelo, una vez que el MGC A ha creado la conexión en el *gateway* A (orden CRCX), le envía un mensaje SIP INVITE al otro MGC incluyendo la descripción de sesión SDP que ha recibido del gateway A. El MGC B crea la conexión en el *gateway* B mediante su correspondiente orden CRCX y tras recibir la respuesta, le envía al MGC A un mensaje SIP 200 OK con la descripción de la sesión. El MGC A reenvía esta información al gateway A en una orden MDCX. Para terminar, MGC A retransmite al MGC B la respuesta positiva a esta orden mediante un mensaje SIP ACK, completándose de esta forma la secuencia de señalización MGCP y SIP. A continuación vendría la transferencia de información multimedia.

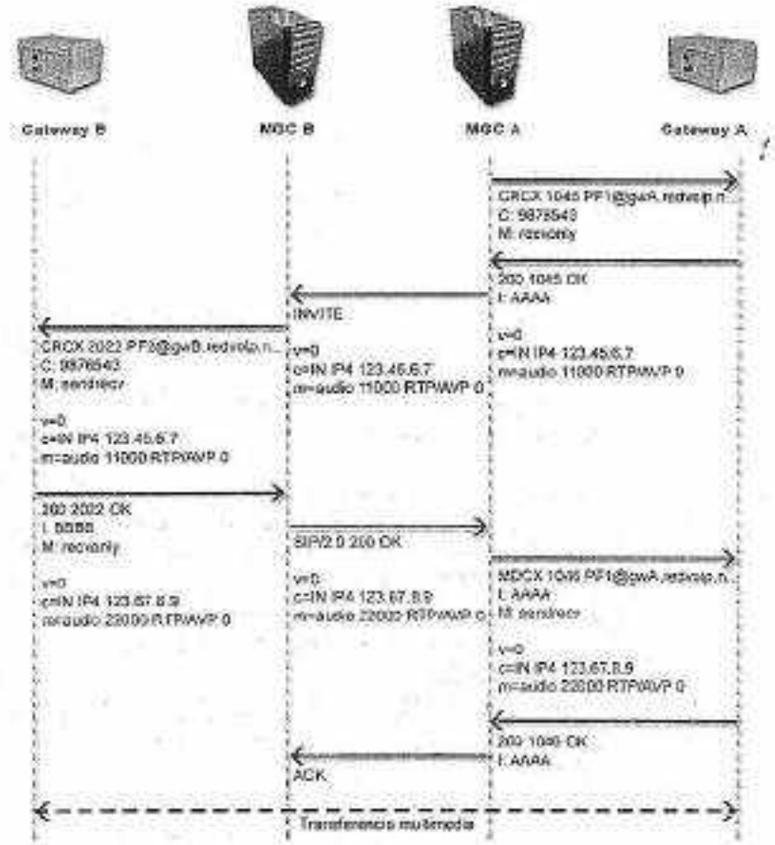

Figura 8.6. Iniciar una llamada con dos MGC

MEGACO/H.248

El IETF creó en enero de 1999 un grupo de trabajo, conocido como Megaco, con el objetivo de crear un protocolo de comunicación entre los MGC y los MG (*gateways*). Este grupo adoptó inicialmente el protocolo MGCP, pero cambió de idea al conocer los trabajos de la UIT-T sobre este mismo tema. La UIT-T tenía al grupo de trabajo SG16 trabajando en el protocolo que primero se conoció como MDCP y posteriormente como H.GCP (*Gateway Control Protocol*, "Protocolo de control del *gateway*"). El resultado fue que ambos organismos reguladores acercaron posturas para obtener como resultado un

estándar común conocido como Megaco o RFC3015 (noviembre de 2000) en el entorno del IETF y como H.248 (junio de 2000) en el entorno de la UIT-T.

El protocolo Megaco tiene parecido con su predecesor MGCP. De hecho, muchos de sus conceptos son similares, aunque la terminología utilizada es distinta. No obstante, mientras que el IETF ha dejado de dar soporte al protocolo MGCP, por lo que cualquier desarrollo futuro será específico de cada proveedor, Megaco ofrece una mayor flexibilidad y dispone de mejores mecanismos para la incorporación de nuevos servicios de valor añadido, lo que hace que tenga unas mejores perspectivas en el mercado.

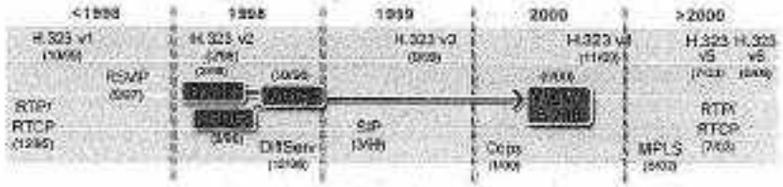

Figura 8.7. El entorno regulatorio de Megaco

El modelo Megaco

En el modelo Megaco/H.248, un *gateway* tiene definidas las siguientes entidades:

- **Terminaciones** (*terminations*). Se trata de entidades emisoras y/o destinatarias de información multimedia. Las terminaciones pueden ser físicas o efímeras. Las primeras tienen una existencia permanente en el *gateway* (por ejemplo, una línea telefónica analógica, un enlace, etc.), mientras que las segundas sólo existen temporalmente (por ejemplo, mientras dura la comunicación, este es el caso de las sesiones RTP). Cada terminación tiene su identificación (*TerminationID*), pudiendo utilizarse los caracteres comodín (* hace referencia a todas y $ hace referencia a la que elija el *gateway*). Por otro lado, cada tipo de terminación tiene sus propias propiedades que se definen por medio de descriptores (*descriptors*).
- **Contextos** (*contexts*). Se trata de una asociación entre dos o más terminaciones. Esta asociación permite el intercambio de información multimedia entre sus terminales. Por tanto, el establecimiento de una comunicación supone añadir dos o más terminaciones a un contexto. Por otro lado, una comunicación se termina eliminando las terminaciones del contexto.

Una terminación sólo puede estar presente en un momento dado en un único contexto. No obstante, una terminación puede cambiarse de un contexto a otro (por

ejemplo para atender una llamada en espera o realizar una consulta). Sólo pueden intercambiar información multimedia las terminaciones que comparten un mismo contexto, aunque, no necesariamente todas ellas tienen que poder emitir y recibir.

Los contextos tienen atributos donde uno de ellos indica el flujo de información de cada terminación. El *gateway* le asigna una identificación a cada contexto (*ContextID*). Existe un contexto particular, conocido como contexto nulo (*null*), al que están asignadas todas las terminaciones que no existen en ningún otro contexto (todas las terminaciones libres). La identificación para este contexto es un guión (-).

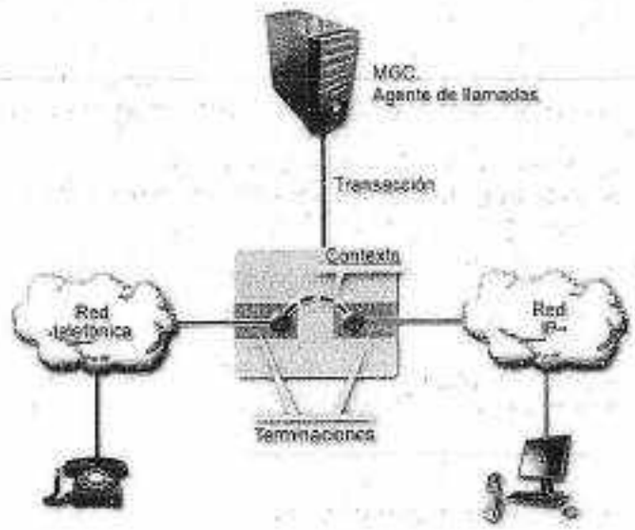

Figura 8.8. El modelo de Megaco

Las transacciones

Los MGC controlan la actividad de los *gateways* mediante el intercambio de órdenes y respuestas conocidas como **transacciones** (*Transactions*).

Una transacción supone el envío de una orden (una *TransactionRequest* o Solicitud de transacción), a la que le seguirá la recepción de una respuesta (una *TransactionReply* o Respuesta de transacción). Generalmente, el MGC le envía la orden al *gateway*, quien, después de ejecutarla, le devuelve la correspondiente respuesta. No obstante, en ciertos casos, este intercambio es en sentido contrario.

Las transacciones siempre hacen referencia a una o más terminaciones de un contexto, por lo que cada transacción debe incluir el *ContextID* y los *TerminationID* afectados.

De acuerdo al protocolo Megaco, las informaciones intercambiadas entre el MGC y el *gateway* pueden estar escritas utilizando un código de texto o un código binario. En el caso del código de texto se utiliza el formato ABNF (*Augmented Backus-Naus Form*, RFC2234), mientras que para el código binario se utiliza ASN.1 (*Abstract Syntax Notation One*, UIT-T X.680). Los MGC deben admitir ambos sistemas para poderse comunicar con el sistema elegido por el *gateway*. En cualquier caso, la única diferencia entre uno y otro sistema es la forma como se representan las informaciones (órdenes, descriptores, etc.) siendo idéntico el modelo de funcionamiento.

Las transacciones se intercambian dentro de un mensaje. En la sintaxis ABNF, los mensajes comienzan con la palabra MEGACO/1, donde 1 es la versión del protocolo. A continuación se incluye la identificación del mensaje (*mID*) y el cuerpo del mismo que contiene las órdenes o respuestas correspondientes. La identificación del mensaje puede ser el nombre de dominio o el número IP de la entidad que lo transmite. Opcionalmente, también se puede incluir el número de puerto.

Las órdenes pueden agruparse por contextos, pudiéndose incluir varios grupos de órdenes de distintos contextos en una misma transacción. Igualmente, un mismo mensaje puede contener más de una transacción:

MEGACO/1 [100.110.120.130]:34567
TransactionRequest (TransactionID1 {
ContextID1 {orden1, orden2, orden3, ...},
ContextID2 {orden1, orden2, orden3, ...},
ContextID3 {orden1, orden2, orden3, ...} })
TransactionRequest (TransactionID2 {
ContextID1 {orden1, orden2, orden3, ...},
ContextID2 {orden1, orden2, orden3, ...},
ContextID3 {orden1, orden2, orden3, ...} })

Cada transacción que se incluye en el mensaje se ejecuta de forma independiente (incluso en orden distinto al especificado). Sin embargo, las órdenes incluidas en una transacción se ejecutan de forma secuencial. De hecho, si por cualquier circunstancia, falla la ejecución de una orden, las siguientes no se llegan a procesar. Este hecho puede evitarse indicando que la ejecución de la orden es opcional. Si una orden opcional falla se pasa a ejecutar la siguiente. Para indicar que la orden es opcional se insertan los caracteres *O-* delante del nombre de la orden.

Una vez que el *gateway* ejecuta todas las órdenes de una transacción devuelve un mensaje de respuesta (*TransactionReply*) con una estructura similar a la de la orden correspondiente. Si se va a tardar un tiempo en terminar la ejecución de la transacción se

puede enviar un *TransactionPending* para evitar que se asuma que se ha perdido la transacción. Esta transacción incluye simplemente el *TransactionID*, sin parámetros.

TransactionPending (TransactionID { })

Las órdenes de Megaco

El protocolo Megaco contempla ocho tipos de órdenes. Salvo *Notify* y *ServiceChange*, el resto de órdenes las envía el MGC al *gateway*. La orden *Notify* la envía el *gateway* al MGC, mientras que *ServiceChange* puede enviarse en ambos sentidos.

Las órdenes son las siguientes:

- ***Add*** (Añadir). Esta orden añade una terminación a un contexto. En el caso de no especificarse el contexto se crea uno nuevo. Por otro lado, si como terminación se utiliza el carácter $ se añadirá una terminación efímera al contexto.
- ***Modify*** (Modificar). Se utiliza para modificar los valores de una terminación, indicarle a una terminación que emita un tono o que detecte e informe de un evento.
- ***Subtract*** (Quitar). Esta orden elimina una terminación de un contexto. Si la terminación eliminada fuese la última de dicho contexto, también se eliminaría éste. En la respuesta a esta orden se puede incluir información estadística de la participación de esta terminación en el contexto. En el caso de una terminación RTP, estas estadísticas incluirían valores como el número de paquetes enviados, recibidos, fluctuación de retardo, etc.
- ***Move*** (Mover). Se utiliza para pasar una terminación de un contexto a otro. Esto hace posible servicios como la llamada en espera.
- ***AuditValue*** (Valor de auditoría). La envía el MGC para conocer los valores actuales de una propiedad, evento o señal asociada con una o más terminaciones.
- ***AuditCapabilities*** (Características de auditoría). La envía el MGC para conocer todos los posibles estados o valores que puede adoptar una propiedad, evento o señal asociada a una o más terminaciones.
- ***Notify*** (Notificar). Esta orden se la envía el *gateway* al MGC para informarle de los eventos que han detectado y que previamente les fueron solicitados (orden *Modify*). Esta orden incluirá el parámetro *RequestID* de la orden en la que se solicitó la notificación.
- ***ServiceChange*** (Cambio de servicio). Cuando es el *gateway* quien envía esta orden al MGC es para indicarle que una terminación deja de estar en servicio o vuelve a estarlo. Cuando es el MGC quien la envía es para indicarle al *gateway* que le pasa el control a otro MGC.

Descriptores de los parámetros

En las transacciones de órdenes y respuestas se envían adicionalmente los valores de los parámetros necesarios para su completa definición. El protocolo Megaco denomina a estos valores como descriptores (en la idea de que terminan de describir la transacción). Cada orden o respuesta tiene su propia lista de descriptores obligatorios, opcionales y prohibidos. El formato general de un descriptor es el siguiente:

Descriptor=<ID>{parámetro=valor, parámetro=valor, ...}

Un descriptor puede incluir uno o más parámetros, estando cada uno de ellos definidos por un valor concreto, un valor que incluye un comodín o una lista de valores en orden de preferencia para que el que lo recibe elija el valor que estime oportuno. El valor elegido lo incluirá en la respuesta. A estas tres formas de definir un parámetro se las conoce como 'especificado completamente' (*fully specified*), 'especificado pobremente' (*underspecified*), o 'especificado en exceso' (*overspecified*).

La lista de descriptores es la siguiente:

- ***Modem Descriptor***. En el caso de que la terminación utilice un módem para la conexión de audio, video o dato, este descriptor define el tipo de módem utilizado y sus parámetros asociados. Se admiten desde los antiguos módem V.22 de 1.200 bps hasta RDSI. No obstante, no es habitual que una terminación tenga conectado un módem de línea telefónica (analógica o RDSI).
- ***Multiplex Descriptor***. Define el tipo de multiplex utilizado en el portador. Se admiten los siguientes tipos: H.221, H.223, H.226, V.76 y Nx64K.
- ***Media Descriptor***. Se trata de un conjunto de descriptores utilizados para definir los diferentes flujos multimedia. Estos descriptores se colocan de forma jerárquica en el siguiente orden:
  Termination State (estado de la terminación). Define los parámetros aplicables a toda la terminación. Contiene las propiedades *Service States* y *Event Buffer Control*. La primera indica si la terminación está disponible para su uso. Puede tomar los valores *test*, *out of service* y *in service*. La segunda especifica si el evento detectado por la terminación debe procesarse inmediatamente (valor *off*) o va a ser guardado en el búfer (valor *lockstep*). Este descriptor le permite al MGC activar o desactivar la notificación inmediata de eventos.
  Stream Descriptor (descriptor de flujo). Se utiliza para identificar los flujos multimedia que están interconectados. En un determinado contexto, los flujos con el mismo *StreamID* están interconectados.
  LocalControlDescriptor (descriptor de control local). Contiene las propiedades específicas del flujo multimedia; particularmente las propiedades *Mode* (*sendonly*, *receiveonly*, *sendreceive*, *inactive* o *loopback*), *ReserveGroup* y *ReserveValue*.
  Local y *RemoteDescriptor* (descriptor local y remoto) contienen la descripción SDP de la sesión para el extremo local y remoto de la conexión.

- ***Events Descriptor***. Contiene un identificador de solicitud (*RequestIdentifier*) y la lista de eventos que debe detectar y notificar el *gateway*. El identificador de solicitud permite relacionar la solicitud con la notificación correspondiente.
- ***SignalsDescriptor***. Contiene la lista de tonos/señales que debe aplicar la terminación. Estas señales pueden aplicarse a un flujo concreto de la terminación o a todos ellos. Existen tres tipos de señales: *on/off* (la señal permanece *on* hasta que específicamente se le indica *off*), *Timeout* (la señal permanece durante un tiempo determinado o hasta que se le indica *off*) y *Brief* (la señal se aplica durante un período de tiempo muy limitado, por ejemplo, un tono MF). El *SignalsDescriptor* puede incluir una secuencia completa de señales.
- ***Audit Descriptor***. Detalla una lista de información que el *gateway* tiene que enviar al MGC. Realmente se trata de los valores de una lista de descriptores que deben incluirse en una respuesta. Estos descriptores son los siguientes: *multiplex*, *events*, *signals*, *observedEvents*, *DigitMap*, *statistics*, *packages* y *EventBuffer*.

CÓDIGO	DESCRIPCIÓN
900	*Servicio restablecido*
901	*Reinicio en frío*
902	*Reinicio en caliente*
903	*Cambio dirigido-MGC*
904	*Mal funcionamiento de la terminación*
905	*Terminación desconectada*
906	*Pérdida de conectividad de capas inferiores*
907	*Fallo de transmisión*
908	*Fallo inminente del gateway*
909	*Fallo inminente del MGC*
910	*Fallo multimedia*
911	*Fallo del módem*
912	*Fallo del multiplexor*
913	*Fallo de señal*
914	*Fallo del evento*
915	*Pérdida de estado*
916	*Cambio de paquete*
917	*Cambio de característica*

Tabla 8.6. Códigos del parámetro ServiceChangeReason

- ***ServiceChange Descriptor***. Se utiliza conjuntamente con la orden *ServiceChange* para incluir informaciones tales como el tipo de cambio de servicio que ha ocurrido o va a ocurrir, la razón de este cambio, y la nueva dirección a utilizar después del cambio. El tipo de cambio de servicio lo define el parámetro *ServiceChangeMethod*, el cual puede tomar uno de los siguientes valores: *graceful* (terminación del servicio sin interrumpir la conexión actual), *forced* (terminación inmediata del servicio), *restart* (restauración del servicio después de un tiempo determinado), *disconnected* (indica que la conexión con el

MGC se ha restablecido después de haber estado interrumpida), *handoff* (indica un cambio de MGC) o *failover* (indica un cambio de *gateway*). El motivo del cambio se incluye en el parámetro *ServiceChangeReason*.

- ***DigitMap Descriptor***. Se trata de la descripción de un plan de numeración. Indica las combinaciones de dígitos que puede marcar el usuario. Este plan de numeración lo guarda el *gateway* para poder enviarle el bloque de dígitos al MGC una vez que el usuario ha terminado de marcarlos y si el número marcado es válido (de otra forma tendría que enviar los dígitos de uno en uno). El plan de numeración se define utilizando una combinación de números, letras y símbolos.

CÓDIGO	DESCRIPCIÓN
400	*Solicitud inválida*
401	*Error de protocolo*
402	*Error de sintaxis en la transacción*
410	*Identificador incorrecto*
411	*La transacción hace referencia a un contexto desconocido*
412	*No hay contextos disponibles*
430	*Identificación de terminación desconocida*
431	*Ninguna terminación se ajusta al comodín*
433	*La terminación ya está en el contexto*
441	*Falta el descriptor remoto*
443	*Orden desconocida*
446	*Parámetro desconocido*
447	*La orden no admite este descriptor*
465	*La transacción inválida no coincide con ninguna solicitud*
500	*Error interno del gateway*
503	*Servicio no disponible*
510	*Recursos insuficientes*
512	*El MG no está equipado para detectar el evento solicitado*
513	*El MG no está equipado para generar la señal solicitada*
515	*Tipo multimedia no admitido*
518	*El búfer de eventos está lleno*
526	*Ancho de banda insuficiente*
530	*Fallo temporal de la red*

Tabla 8.7. Ejemplos de códigos de error del protocolo Megaco

- ***StatisticDescriptor***. Ofrece información estadística del uso de una terminación en un contexto. Las estadísticas devueltas dependen del tipo de terminación en cuestión. Este descriptor se incluye en la respuesta a una orden *Subtract* o *AuditValue*.
- ***ObservedEvents Descriptor***. Relaciona los eventos que ha observado el *gateway*. Este descriptor es obligatorio con la orden *Notify* para informar al MGC de los eventos que se han observado. Cuando se utiliza en la respuesta a una orden *AuditValue* se hace para informar de los eventos del búfer que estaban pendientes de informar al MGC.

- ***Error Descriptor***. Se incluye en una respuesta cuando la orden solicitada no se ha podido ejecutar. Este descriptor está formado por un código de error y, opcionalmente, por un texto de descripción.
- ***Topology Descriptor***. Indica cómo se moverá el flujo multimedia dentro de un contexto (quién puede oír a quién). Por defecto, todas las terminaciones de un contexto pueden enviar y recibir la información multimedia al y del resto de terminaciones. *Topology Descriptor* se utiliza para indicar una circunstancia diferente. El descriptor indica si entre dos terminaciones no debe haber flujo de información multimedia (*isolate*), si sólo se debe dar en un sentido (*oneway*) o en los dos sentidos (*bothway*). Por ejemplo, *T1, T2, oneway* indica que T1 puede enviar información a T2, pero T2 no puede enviar información a T1. Este descriptor es interesante para la implementación de servicios como la llamada en espera o la conferencia a tres. Como este descriptor hace referencia a un contexto, y no a una terminación, cuando se utiliza en una transacción debe incluirse antes de la lista de comandos para las terminaciones particulares. Por otro lado, sólo es aplicable a los contextos existentes.

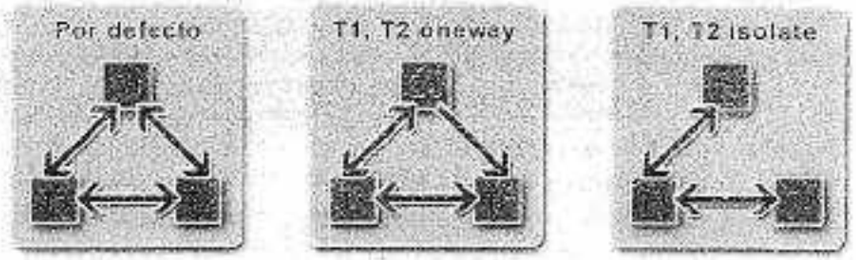

Figura 8.9. Ejemplos de topologías

- ***Packages Descriptor***. Se utiliza exclusivamente como respuesta a una orden *AuditValue* para informar al MGC de los paquetes implementados en una terminación. Los paquetes son grupos de propiedades, señales, eventos y estadísticas relacionadas con las terminaciones. Existen paquetes relacionados con la detección de tonos en la línea, con la generación de tonos, enlaces TDM, etc. Cada tipo de terminación tiene sus características, de forma que implementa distintos tipos de paquetes. Megaco tiene definido un conjunto de paquetes para cubrir la mayoría de las necesidades actuales, no obstante, también describe el procedimiento para crear nuevos paquetes y registrarlos en el IANA. Esto permite, por ejemplo, que existan paquetes para el manejo de líneas o terminales específicos de un fabricante.

PAQUETE	SÍMBOLO
PAQUETE GENÉRICO	*g*
PAQUETE GENERADOR DE TONOS	*tonegen*
PAQUETE DETECCIÓN DE TONOS	*tonedet*
Detectado inicio de tono	*std*
Detectado fin de tono	*etd*
Detectado tono largo	*ltd*
PAQUETE GENERADOR DTMF BÁSICO	*dg*
Dígito DTMF 0-9#*	*d0-9*#
PAQUETE DETECCIÓN DTMF	*dd*
PAQUETE GENERADOR DE TONOS DE LLAMADA	*cg*
Tono de invitación a marcar	*dt*
Tono de llamada	*rt*
Llamada en espera	*cw*
Ocupado	*bt*
PAQUETE TONOS DE PROGRESO DE LLAMADA	*cd*
PAQUETE DE SUPERVISIÓN DE LÍNEA ANALÓGICA	*al*
Descolgado	*of*
Colgado	*on*
PAQUETE DE PRUEBAS DE CONTINUIDAD	*ct*
PAQUETE DE RED	*nt*
PAQUETE RTP	*rtp*
PAQUETE DE CIRCUITO TDM	*tdmc*

Tabla 8.8. Ejemplos de paquetes básicos y eventos (lista no exhaustiva, RFC3015)

Iniciar una llamada

Como ejemplo de la utilización del protocolo Megaco/H.248 vamos a recrear los mensajes que se intercambian el MGC y los *gateways* involucrados en el establecimiento de una llamada. Vamos a suponer que un terminal analógico conectado a la terminación A1111 del *gateway* 1 inicia una llamada a un terminal atendido por el *gateway* 2. Ambos *gateways* son controlados por el mismo MGC. Supondremos que las direcciones IP son 100.1.1.1, 100.2.2.2 y 100.3.3.3 para el *gateway* 1, *gateway* 2 y MGC respectivamente, siendo el puerto utilizado el 55555 en todos los casos.

Los mensajes que se muestran a continuación están simplificados en la idea de hacerlos más simples y que se pueda percibir el proceso con mayor claridad.

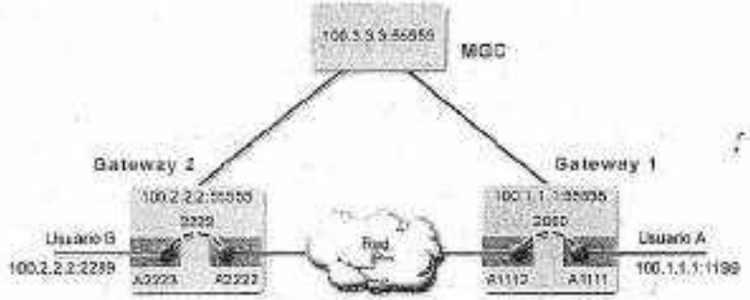

Figura 8.10. La interconexión de redes del ejemplo

Como primer paso, el *gateway* 1 detecta que se ha originado un descolgado en la terminación A1111 (*al/of*) y se lo notifica al MGC, a lo que el MGC le responde con la confirmación correspondiente:

1.- *Gateway* 1 al MGC

```
MEGACO/1 [100.1.1.1]:55555
Transaction = 10000 {
   Context = - {
      Notify = A1111 {ObservedEvents =2222 {
         20060729T22000000:al/of}}
}}
```

2.- MGC al *gateway* 1

```
MEGACO/1 [100.3.3.3]:55555
Reply = 10000 {
   Context = - {Notify = A1111}
}
```

El MGC le ordena a continuación al *gateway* 1 que genere el tono de marcado en la terminación A1111 para invitar a su usuario (usuario A) a que marque el número telefónico con el que desea establecer la comunicación. Cuando el usuario marca el primer dígito, se debe cortar la emisión del tono. El número telefónico que marque el usuario A debe cumplir el plan de numeración *Dialplan0*. Además, ahora el *gateway* tiene que estar pendiente del evento colgado (por si el usuario decide colgar en cualquier momento). El mensaje sería el siguiente:

3.- MGC al *gateway* 1

```
MEGACO/1 [100.3.3.3]:55555
Transaction = 10001 {
   Context = - {
      Modify = A1111{
```

```
        Events = 2223 {
            al/on, dd/ce {DigitMap=Dialplan0}},
        Signals {cg/dt},
        DigitMap= Dialplan0{
(0|00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxx|9011x.)}
}))
```

4.- *Gateway* 1 al MGC

```
MEGACO/1 [100.1.1.1]:55555
Reply = 10001 {
    Context = - {Notify = A1111}
}
```

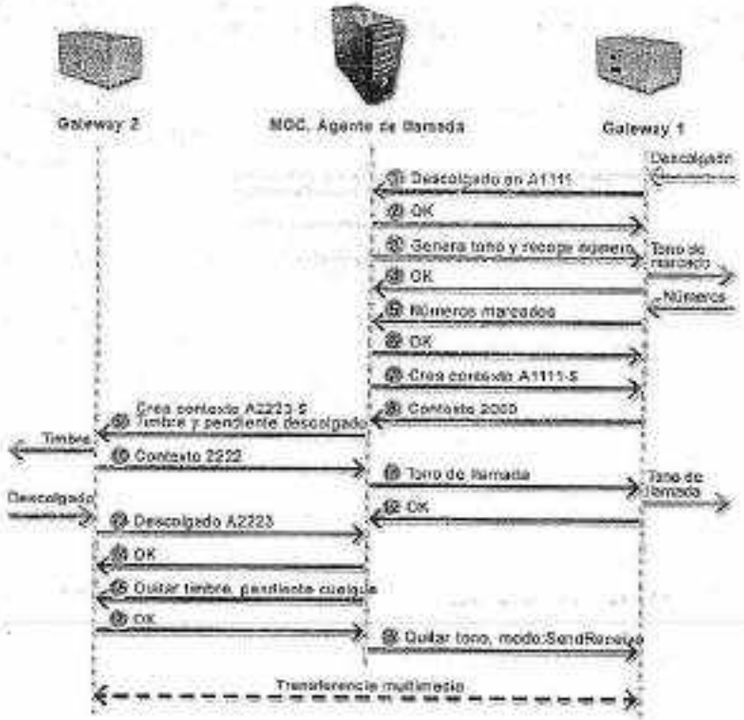

Figura 8.11. Establecimiento de una comunicación con Megaco/H.248

El *gateway* 1 va guardando los dígitos que marca el usuario A, comprueba que cumple el plan de numeración y se los pasa al MGC en un mensaje de notificación. Éste le responde con una confirmación de recepción:

5.- *Gateway* 1 al MGC

```
MEGACO/1 [100.1.1.1]:55555
Transaction = 10002 {
    Context = - {
        Notify = A1111{ObservedEvents=2223}
20060729T22010001:dd/ce{ds="912125552l2",Meth=FM}
}}}
```

6.- MGC al *gateway* 1

```
MEGACO/1 [100.3.3.3]:55555
Reply = 10002 {
    Context = - {Notify = A1111}
}
```

El MGC analiza la numeración y determina que el número marcado se atiende desde el *gateway* 2, por lo que hay que establecer un enlace entre el *gateway* 1 y el *gateway* 2. Para ello, en primer lugar crea un contexto nuevo (*context = $*) en el *gateway* 1 entre la terminación A1111 y una terminación RTP. Al *MediaDescriptor* de la segunda terminación le asigna un determinado identificador de flujo (*stream = 1*), fija, por ahora, el modo de la comunicación de esta terminación en *ReceiveOnly* (sólo recibir), y hace dos sugerencias de codecs: audio G.711 ley mu (RTP tipo 0) y audio G.728 (RTP tipo 15). Como descripción SDP del terminal remoto se incluye el comodín $. El mensaje sería el siguiente:

7.- MGC al *gateway* 1

```
MEGACO/1 [100.3.3.3]:55555
Transaction = 10003 {
    Context = $ {
        Add = A1111,
        Add = $ {
            Media {
                Stream = 1 {
                    LocalControl {
                        Mode = receiveonly}
                    Local {
                        v=0
                        c=IN IP4 $
                        m=audio $ RTP/AVP 0
                        v=0
                        c=IN IP4 $
                        m=audio $ RTP/AVP 15}
}}}}}
```

El *gateway* 1 le asigna al contexto la identificación 2000 y a la nueva terminación le asigna la identificación A1112. De la misma forma, incluye la dirección IP (100.1.1.1) y puerto (1199) del terminal local y se decide por el codec G.711 ley mu (RTP tipo 0):

8.- *Gateway* 1 al MGC

```
MEGACO/2 [100.1.1.1]:55555
Reply = 10003 {
   Context = 2000 {
      Add = A1111,
      Add = A1112 {
         Media {
            Stream = 1 {
               Local {
                  v=0
                  c=IN IP4 100.1.1.1
                  m=audio 1199 RTP/AVP 0}
}}}}}
```

A continuación, el MGC le envía un mensaje al *gateway* 2 para indicarle que cree un nuevo contexto con dos terminaciones: A2223 y $. En este caso, el modo se fija a *SendReceive*, se le sugiere que el codec sea G.711 ley mu (RTP tipo 0), se le informa del número IP y puerto remoto (100.1.1.1:1199), se le indica que haga sonar el timbre del usuario B y que esté pendiente del descolgado:

9.- MGC al *gateway* 2

```
MEGACO/2 [100.3.3.3]:55555
Transaction = 10004 {
   Context = $ {
      Add = A2223 {Media}
         Stream = 1 {
            LocalControl{
               Mode=SendReceive}}},
         Events=1234{al/of},
         Signals{al/ri}},
      Add = $ {
         Media {
            Stream = 1 {
               LocalControl {
                  Mode = sendreceive)
               Local {
                  v=0
                  c=IN IP4 $
                  m=audio $ RTP/AVP 0),
               Remote {
                  v=0
                  c=IN IP4 100.1.1.1
                  m=audio 1199 RTP/AVP 0}
}}}}
```

De forma similar a la respuesta del *gateway* 1, el *gateway* 2 responde con el siguiente mensaje:

10.- *Gateway* 2 al MGC

```
MEGACO/2 [100.2.2.2]:55555
Reply = 10004 {
   Context = 2222 {
      Add = A2223,
      Add = A2222 {
         Media {
            Stream = 1 {
               Local {
                  v=0
                  c=IN IP4 100.2.2.2
                  m=audio 2299 RTP/AVP 0}
}}}}}
```

A continuación, el MGC le envía un nuevo mensaje al *gateway* 1 con una orden de modificación donde le indica el número IP (100.2.2.2) y puerto (2299) de la terminación remota, así como que le haga sonar el tono de confirmación de llamada al usuario A. El *gateway* responde confirmando los cambios:

11.- MGC al *gateway* 1

```
MEGACO/2 [100.3.3.3]:55555
Transaction = 10005 {
   Context = 2000 {
      Modify = A1111{
         Signals{cg/rt}},
      Modify = A1112 {
         Media {
            Stream = 1 {
               LocalControl {
                  Mode = sendreceive}
               Remote {
                  v=0
                  c=IN IP4 100.2.2.2
                  m=audio 2299 RTP/AVP 0}
}}}}}
```

12.- *Gateway* 1 al MGC

```
MEGACO/2 [100.1.1.1]:55555
Reply = 10005 {
   Context = 2000 {
      Modify = A1111, Modify = A1112}
}
```

En este momento está sonando el timbre del terminal del usuario B y el usuario A está recibiendo el tono de llamada correspondiente. Cuando el usuario B descuelga, el *gateway* 2 le informa al MGC, quien le confirma la notificación y le ordena que deje de hacer sonar el timbre del usuario B (*Signals { }*):

13.- *Gateway* 2 al MGC

```
MEGACO/2 [100.2.2.2]:55555
Transaction = 10006 {
   Context = 2222 {
      Notify = A2223 {ObservedEvents =1234 {
         20060729T22020002:al/of}
}}
```

14.- MGC al *gateway* 2

```
MEGACO/2 [100.3.3.3]:55555
Reply = 10006 {
   Context = - {Notify = A2223}
}
```

15.- MGC al *gateway* 2

```
MEGACO/2 [100.3.3.3]:55555
Transaction = 10007 {
   Context = 2222 {
      Modify = A2223 {
         Events = 1235 {al/on},
         Signals {}
}}}
```

16.- *Gateway* 2 al MGC

```
MEGACO/2 [100.2.2.2]:55555
Reply = 10007 {
   Context = 2222 {Modify = A2223}
}
```

Por último, el MGC le indica al *gateway* 1 que cambie el modo a *SendReceive* y deje de hacer sonar el tono de llamada. La comunicación ha quedado establecida:

17.- MGC al *gateway* 1

```
MEGACO/2 [100.3.3.3]:55555
Reply = 10008 {
   Context = 2000{
      Modify = A1112 {
         Media{
            Stream = 1 {
               LocalControl{
                  Mode=SendReceive}}},
      Modify = A1111 {
         Signals {}
}}}
```

Garantizar la calidad de servicio

NECESIDAD DE LA CALIDAD DE SERVICIO

Las redes IP hacen lo posible por gestionar absolutamente todo el tráfico que reciben de sus usuarios. Cuando este tráfico se mantiene dentro de unos límites, los paquetes van pasando de nodo en nodo hasta llegar a su destino sin sufrir mayores inconvenientes. No obstante, conforme el tráfico aumenta, determinados dispositivos o enlaces de la red empiezan a estar saturados, con lo que los paquetes se van almacenando en memorias intermedias (búferes) para irlos gestionando poco a poco conforme se van liberando los recursos. Si se llegan a saturar las memorias intermedias, los nuevos paquetes que llegan se desechan directamente.

Este modelo de red basado en hacer todo lo posible (lo que en inglés se conoce como *best effort network*) tiene el inconveniente de no poder garantizar que los paquetes vayan a llegar a su destino o el tiempo que van a tardar en hacerlo. Por otro lado, como las cargas de trabajo se producen de forma aleatoria en distintos dispositivos de la red, los paquetes de una misma sesión pueden sufrir retardos distintos, dando lugar al *jitter* o fluctuación de retardo.

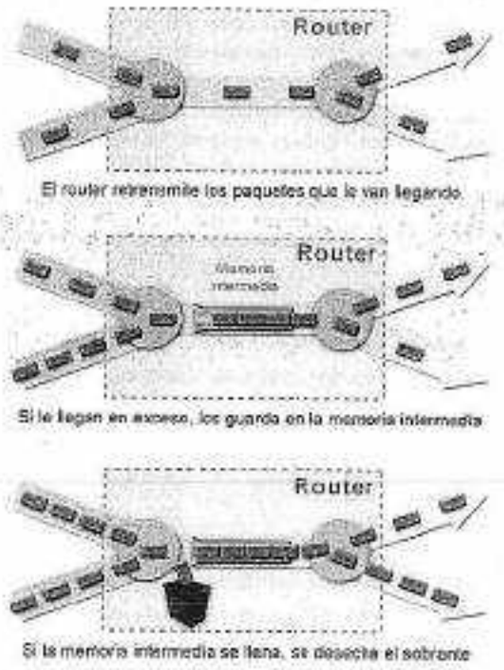
Router
El router retransmite los paquetes que le van llegando
Router
Memoria intermedia
Si le llegan en exceso, los guarda en la memoria intermedia
Router
Si la memoria intermedia se llena, se desecha el sobrante

Figura 9.1. Funcionamiento del router

Ciertamente, si los paquetes que sufren retraso corresponden a un correo electrónico, a una página Web o a una transferencia de archivo, estos milisegundos adicionales que les lleva alcanzar su destino no suponen una merma del servicio. Por otro lado, para evitar la pérdida de paquetes se diseñó el protocolo TCP. Con este protocolo se detecta dicha pérdida y se reenvía el paquete; eso sí, a cambio de cierto retardo. Como se puede suponer, el problema se produce cuando estos paquetes pertenecen a aplicaciones de tiempo real (transmisiones de voz y vídeo fundamentalmente). En estos casos, un retardo adicional de unos milisegundos o la fluctuación del retardo de los distintos paquetes, puede degradar significativamente la calidad de servicio. Esto quiere decir que, en estos casos, la voz o el vídeo se escucharán o se verá con peor calidad.

APLICACIÓN	ANCHO DE BANDA	RETARDO	FLUCTUACIÓN DE RETARDO	ERRORES O PÉRDIDAS
Correo electrónico	*Baja*	*Baja*	*Baja*	*Alta*
Transferencia de archivo	*Alta*	*Baja*	*Baja*	*Alta*
Navegación Web	*Mediana*	*Mediana*	*Mediana*	*Alta*
Datos interactivos	*Baja*	*Baja*	*Baja*	*Alta*
Difusión de audio	*Mediana*	*Baja*	*Alta*	*Mediana*
Voz	*Mediana*	*Mediana*	*Alta*	*Baja*
Difusión de vídeo	*Alta*	*Baja*	*Alta*	*Mediana*
Vídeo y audio interactivo	*Alta*	*Mediana*	*Alta*	*Mediana*

Tabla 9.1. Necesidades de los distintos tipos de tráfico de las redes IP

Existen dos alternativas para resolver el problema de la congestión de la red:

- **Sobredimensionar la red**. Si todos los enlaces de la red dispusieran de un ancho de banda mucho mayor del necesario, aún en las peores condiciones de tráfico, la red nunca llegaría a estar congestionada. Todo el tráfico que llega a un dispositivo sería reenviado inmediatamente por su ruta correspondiente, sin retardo. El inconveniente de esta solución es que para asegurar que una ruta no va a sufrir congestión, aún en las peores condiciones, habría que multiplicar su capacidad por 10 o por 100, con el consiguiente incremento de coste que esto supone. Una forma de sobredimensionar la red es utilizar redes separadas para cada tipo de tráfico, dimensionada cada una de ellas de acuerdo a sus propias necesidades. En cualquier caso, sobredimensionar no cabe duda de que es una solución, pero excesivamente cara.
- **Técnicas de calidad de servicio**. Si determinadas aplicaciones no son sensibles al retardo o fluctuaciones de retardo (correo electrónico, Web o transferencia de archivos), mientras que otras sí lo son (voz y vídeo), si se diferenciara el tráfico de cada una de ellas, la red podría ofrecerles un tratamiento distinto, satisfaciendo de esta forma las distintas necesidades sin aumentar considerablemente el coste de red. A este sistema se lo conoce como de calidad de servicio o QoS (*Quality of Service*).

Por tanto, las técnicas de QoS son la mejor forma de adaptar una red IP para que pueda ser utilizada por todo tipo de servicios. No obstante, salvo en el caso de redes IP privadas, la QoS tiene un inconveniente: Internet está formada por múltiples redes interconectadas sin garantía de que todas ellas apliquen las mismas soluciones de QoS. En el caso de las redes IP privadas, como por ejemplo una red corporativa de una empresa o la de un proveedor ISP, todos los nodos pueden tener instalado un mismo sistema de QoS y, por tanto, no tiene que haber problemas de falta de homogeneidad.

Figura 9.2. Efecto de la calidad de servicio

La red telefónica tradicional no tiene problemas de calidad de servicio, ya que reserva un circuito extremo a extremo para cada comunicación (conmutación de circuitos). Por otro lado, en el caso de que la red detecte una saturación de cualquiera de sus recursos, simplemente bloquea el establecimiento de la comunicación. Esto quiere decir que la red telefónica tradicional no establece la comunicación a menos que tenga garantías de calidad; o lo que es lo mismo, la red telefónica es una red de calidad garantizada frente al modelo de 'hare todo lo posible' de las redes IP (conmutación de paquetes). También es cierto que los servicios de estas últimas son mucho más económicos que los de la primera.

Otra pregunta distinta sería cómo se mide la calidad de servicio. Desde el punto de vista general se podría decir que es una medida del cumplimiento de las expectativas del cliente. Llevando esto a la voz, básicamente, el cliente espera que el establecimiento de la llamada se produzca de forma rápida y poder disfrutar de una buena calidad de sonido durante el transcurso de la comunicación. Si trasladamos esto a las redes IP estaríamos hablando en términos de ancho de banda, pérdida de paquetes, retardo y fluctuaciones de retardo (*jitter*).

Conviene destacar que cuando se habla de QoS se está haciendo referencia a las prestaciones de la red sin tener en cuenta la influencia que tienen las distintas técnicas de codificación, protocolos de transporte (RTP/RTCP) o de señalización (SIP o H323) en la

calidad final del servicio que se ofrece al usuario. Tampoco se consideran las aplicaciones de usuario o los terminales utilizados.

La calidad más allá de la técnica

Un usuario de un servicio de telefonía IP no tiene por qué conocer los detalles de cómo un operador resuelve su llamada, lo que le importa es establecer su comunicación con comodidad y mantener una conversación con calidad de voz. Lo cierto es que ese es también el objetivo del operador del servicio. Sin embargo, en Internet ocurre que una llamada telefónica puede iniciarse en la red de un operador y terminarse en la de otro, pudiendo atravesar distintas redes intermedias. Basta con que una de las redes no trabaje orientada a la calidad para que el servicio que perciba el usuario sea deficiente.

Hay que tener en cuenta que no todas las redes que forman Internet se gestionan de la misma forma o de acuerdo con los mismos criterios. Por tanto, la única forma de garantizar la calidad del servicio es haciendo que la comunicación se establezca dentro de una misma red o entre redes que trabajan de una forma conjunta en estos temas.

Para hacer que dos o más redes trabajen de una forma coordinada en relación con la calidad de servicio se deben definir primero las necesidades de cada una de ellas y determinar las soluciones técnicas correspondientes. A continuación se crea un documento que recoja lo que cada red (cada operador) debe esperar de la otra y lo que debe ofrecerle. Esto se conoce como SLA (*Service Level Agreements*, 'Acuerdos de nivel de servicio'). Por tanto, un SLA es un acuerdo entre operadores que recoge sus compromisos en cuanto a la calidad de servicio, así como, las penalizaciones correspondientes si no se cumple lo pactado.

Lo cierto es que, hasta que todos los proveedores de Internet participen en estos acuerdos de calidad de servicio, no se podrá garantizar unas comunicaciones de calidad entre cualquier punto de la red, como ocurre actualmente con la telefonía tradicional. Por este motivo, los operadores suelen diferenciar en sus contratos la calidad ofertada dentro de su red y con el resto de Internet.

Por otro lado, la calidad de servicio es algo más que una buena calidad de voz. En un entorno tan competitivo como Internet, uno de los grandes aportes de esta red es su gran potencial para añadirle nuevos servicios a la voz. Por tanto, también en estos aspectos hay que satisfacer a los clientes, además de en los aspectos tradicionales de un buen servicio de atención al cliente, una implementación rápida y eficaz, una facturación eficiente, etc.

TÉCNICAS DE CALIDAD DE SERVICIO

Como se ha visto, las técnicas de QoS consisten en diferenciar cada tipo de tráfico para poderles ofrecer un servicio diferente. Estas técnicas se implementan en los nodos de red (*routers* y *switches*) formando lo que se conoce como arquitectura de calidad de servicio.

Las arquitecturas más populares son:

- **Servicios diferenciados o DiffServ** (*Differentiated Services*). En este caso, se marca cada paquete con el tipo de QoS adecuado dependiendo de determinados parámetros (tipo de información que transporta, protocolo utilizado, etc.). Para ello se utiliza el campo *ToS* en IPv4 o *clase de tráfico* en IPv6. Los nodos de la red que sean compatibles con este sistema utilizan este valor para clasificar el tráfico y gestionarlo con una prioridad distinta. Las principales ventajas de este sistema de QoS son: su adopción es simple, basta con marcar los paquetes; los dispositivos de red no tienen que asociar los paquetes como pertenecientes a un mismo flujo de tráfico, sino que simplemente le van aplicando distinta prioridad conforme van llegando; y, por último, no se requiere un protocolo específico que gestione la calidad de servicio, por lo que el flujo de información es más ligero. Su principal inconveniente es que este sistema no dispone de control de admisión, por lo que, realmente, no ofrece garantías de calidad de servicio, sino garantías de prioridad.
- **Reserva de recursos o IntServ** (*Integrated Services*, Servicios Integrados). En este caso se utiliza un protocolo específico para reservar los recursos de red necesarios que aseguren la calidad de servicio esperada. Los paquetes se identifican como pertenecientes a un mismo flujo, aplicándoseles el tratamiento previamente acordado. La mayor ventaja de este sistema es que permite asegurar la calidad de servicio (dispone de control de admisión), existiendo, además, una coordinación entre las decisiones de QoS de la red y las aplicaciones. Por otro lado, su mayor inconveniente es que resulta más complejo de implantar y requiere disponer de mayores recursos (mayor capacidad de proceso de los nodos, además de que el flujo resulta algo más pesado al utilizarse un protocolo adicional). IntServ utiliza el protocolo RSVP (*Resource ReserVation Protocol*, Protocolo de reserva de recursos).
- **Juntar IntServ y DiffServ**. Dado que la mayor ventaja de IntServ es poder garantizar la QoS, mientras que requiere mayores recursos, y la mayor ventaja de DiffServ es su simplicidad mientras que no garantiza la QoS, el IETF sugirió, en su recomendación RFC2998, combinar ambos servicios para sumar sus ventajas. En esta recomendación se describe cómo ofertar un servicio IntServ sobre una red DiffServ. La idea consiste en disponer de regiones DiffServ dentro de la comunicación IntServ extremo a extremo. Cuando los paquetes viajan por los nodos no compatibles con RSVP utilizan DiffServ. Los mensajes RSVP se utilizan de nuevo en las regiones IntServ. Esta solución hace posible utilizar IntServ en redes que no pueden garantizar que todos sus nodos lo admitan (Internet).

Es interesante mencionar que, hoy por hoy, la única técnica que puede garantizar la calidad ofertada es IntServ, quien, curiosamente, hace que la red IP se comporte como una red de conmutación de circuitos, asegurando que los recursos están disponibles antes de establecer la comunicación. Por tanto, alguien podría preguntarse por qué no se sigue utilizando la red telefónica tradicional para la voz, dejando la red IP para los datos. La respuesta es que la red IP no sólo es de menor coste, sino que también es más versátil a la hora de incorporar nuevos servicios o integrar varios de ellos. Dicho de otra forma, ofrece nuevas posibilidades.

PROPIEDAD	SERVICIOS INTEGRADOS	SERVICIOS DIFERENCIADOS
Aplicación	*Por flujo (extremo a extremo)*	*Por paquetes (por salto)*
Clasificación del tráfico	*Protocolo de señalización*	*Campo DS en la cabecera IP*
Control de admisión	*Sí*	*No*
Protocolo de señalización propio	*Sí*	*No*
Escalabilidad	*Limitada por el número de flujo*	*Limitada por el número de clases*
Contabilidad de la red	*Basada en las necesidades del flujo*	*Basada en el uso de las clases*
Gestión de la red	*Similar al de las redes de conmutación de circuitos*	*Similar al de las redes actuales de IP*
Desarrollo entre dominios	*Acuerdos multilaterales*	*Acuerdos bilaterales*
Garantía de QoS	*Sí*	*No*
Implantación	*Complejo*	*Simple*

Tabla 9.2. Ejemplos de RFC sobre el modelo de servicios diferenciados

SERVICIOS DIFERENCIADOS

El modelo de servicios diferenciados o DiffServ se basa en marcar cada paquete con una prioridad distinta dependiendo del tipo de tráfico del que forma parte. Posteriormente, los nodos de la red sólo tienen que aplicarle la prioridad correspondiente a cada uno. La valoración de la prioridad que se merece cada paquete puede llevarse a cabo teniendo en cuenta distintos parámetros, como pueden ser: protocolo utilizado, tipo de información que transporta, etc.

En cualquier caso, la implantación de este modelo es relativamente simple, ya que no requiere incorporar un protocolo de enrutamiento adicional, ni supone utilizar recursos extraordinarios en los nodos.

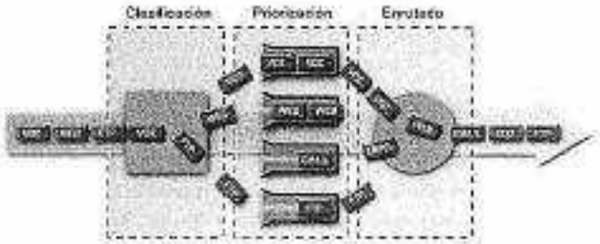

Figura 9.3. Diferenciación de servicios

Como a cada paquete se le va comprobando su prioridad de nodo en nodo, al sistema se le ha venido a llamar PHB (*Per Hop Behaviour*, Comportamiento por salto).

Por cierto, el protocolo DiffServ se describe en la recomendación RFC2475, titulada '*An Architecture for Differentiated Services*' (Una arquitectura de servicios diferenciados).

RFC	FECHA	DESCRIPCIÓN
RFC2430	10/1998	Arquitectura del proveedor de servicios DiffServ
RFC2474	12/1998	Definición del campo DS en las cabeceras IPv4 y IPv6
RFC2475	12/1998	Arquitectura de los servicios diferenciados
RFC2597	06/1999	Servicio Expedited Forwarding
RFC2598	06/1999	Servicio Assured Forwarding
RFC2638	07/1999	Arquitectura DiffServ de dos bits
RFC2697	09/1999	Marcador de tres colores con tasa única
RFC2698	09/1999	Marcador de tres colores con tasa doble
RFC2963	10/2000	Adaptador de velocidad (shaper) para DiffServ
RFC2983	10/2000	DiffServ y tunelado
RFC3086	04/2001	Definición de DiffServ
RFC3246	03/2002	Servicio Expedited Forwarding
RFC3270	05/2002	MPLS para DiffServ
RFC3287	07/2002	Seguimiento remoto de DiffServ
RFC3289	05/2002	MIB para la arquitectura DiffServ
RFC4115	07/2005	Marcador de tres colores con tasa doble

Tabla 9.3. Ejemplos de RFC sobre el modelo de servicios diferenciados

El reto que presenta esta tecnología es determinar quién clasifica los paquetes y qué tipo de prioridad se le da a cada tipo de tráfico. En principio, cada paquete podría venir ya marcado desde el terminal origen, pero esto tiene el inconveniente de que cada terminal (cada aplicación) podría aplicar distintas reglas de clasificación del tráfico o, incluso, podría haber aplicaciones que marcaran sus paquetes con una prioridad superior a la que le corresponde para conseguir mejorar su eficiencia a costa de empeorar la de

otros. El hecho es que para que el sistema DiffServ funcione es necesario contar con un sistema racional de clasificación del tráfico y un mecanismo objetivo de marcado del mismo.

El dominio DS

En la idea de simplificar el proceso de clasificación y marcado de los paquetes, el IETF estableció lo que se conoce como dominio de servicios diferenciados o dominio DS. Un dominio DS es un conjunto de nodos que utilizan una misma política de clasificación y marcado de paquetes. Esto quiere decir que una vez que el primer nodo marca el paquete, el resto de nodos sólo tiene que tratarlo de acuerdo a la prioridad marcada, sin más análisis.

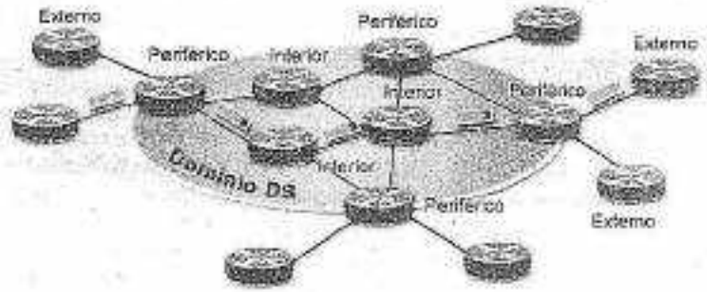

Figura 9.4. El dominio DS

Un dominio DS lo suelen formar los nodos de un mismo proveedor, siendo su administrador de red quien fija las políticas de clasificación de paquetes correspondientes. Como cada paquete se marca a la entrada del dominio, dos o más redes interconectadas que utilicen las mismas políticas pueden formar un único dominio, evitando, de esta forma, tener que clasificar y marcar cada paquete de forma reiterada. Las políticas de clasificación y marcado de paquetes a aplicar en estas redes se acuerdan conjuntamente y quedan reflejadas en un SLA. Este SLA tiene un elemento importante que se conoce como TCA (*Traffic Conditioning Agreement*, 'Acuerdo de clasificación del tráfico'), el cual establece las reglas de clasificación del tráfico, así como las mediciones, marcado, descartes y/o reglas de conformación aplicables.

Por tanto, un dominio DS dispone de dos tipos de nodos:

- **Nodos periféricos** (*boundary nodes* o *edge nodes*). Se trata de los nodos que delimitan el dominio, los que son frontera entre el exterior y el propio dominio.

Estos nodos se encargan de marcar los paquetes que entran en el dominio, de forma que los nodos internos no tengan que perder tiempo en volverlos a evaluar, sino que simplemente aplican la prioridad definida para cada tipo de tráfico. Cuando los paquetes salen del dominio, el nodo periférico correspondiente, los retoca para adaptarlos a los requerimientos del nuevo dominio en el que entran. Por tanto, los nodos periféricos pueden dividirse en nodos de entrada (*ingress nodes*) y nodos de salida (*egress nodes*).

- **Nodos interiores** (*interior nodes* o *core nodes*). Se trata de nodos que están interconectados exclusivamente con otros nodos del dominio (sean interiores o periféricos). Estos nodos se limitan a aplicar las reglas de priorización adecuadas a cada tipo de tráfico.

Marcado de los paquetes

Para marcar los paquetes con el tipo de prioridad correspondiente se utiliza el campo ToS (*Type of Service*, 'Tipo de servicio') de IPv4 o el campo clase de tráfico (*Traffic Class*) de IPv6. Aunque estos campos tienen ocho bits, el sistema DiffServ sólo utiliza los primeros 6 bits, a los que les asigna el nombre de campo DS (*Differentiated Services*, 'Servicios diferenciados'). Los otros dos bits, conocidos como ECN (*Explicit Congestion Notification*, 'Notificación explícita de la congestión'), se utilizan actualmente de forma experimental para mecanismos de control de congestión. Por cierto, al tipo de prioridad se lo conoce también como tipo de servicio o tipo de comportamiento PHB.

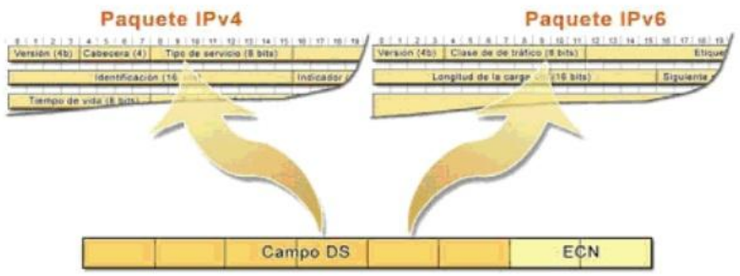

Figura 9.5. El campo DS en los paquetes IP

Para fijar la prioridad se utiliza un código numérico, conocido como DSCP (*Differentiated Services Codepoint*, 'Código de servicios diferenciados'). En general, cuanto mayor es el valor de este código, mayor es la prioridad asignada al paquete.

Los 64 códigos posibles (6 bits) están divididos en tres grupos:

- 32 códigos (xxxxx0). Utilizados para asignar la prioridad o comportamiento a cada uno de los paquetes.
- 16 códigos (xxxx11). Utilizados para experimentaciones o de forma local.
- 16 códigos (xxxx01). Para ser utilizados en el caso de que el primer grupo de 32 códigos se quede pequeño.

Clasificación del tráfico

La clasificación del tráfico puede llevarse a cabo en función de distintos parámetros, como por ejemplo: el protocolo utilizado, el tipo de tráfico, el origen o destino, etc. Los parámetros utilizados y valoración de los mismos dependen de cada administrador, lo que sí define el IETF son tres tipos de comportamientos o prioridades:

- **Reenvío urgente** o **EF** (*Expedited Forwarding*, RFC3246). Se trata del nivel de prioridad máximo, marcado con el código DSCP 101110. Los nodos del dominio DS correspondiente tratarán los paquetes EF de forma que le aseguren un nivel bajo de pérdidas, retardo y fluctuación de retardo (*jitter*). A este comportamiento se lo conoce también como premium y, aunque no ofrece garantías cuantitativas para la tasa de pérdidas, retardo o fluctuación de retardo, intenta ofrecer un servicio similar al de un circuito punto a punto.
- **Reenvío asegurado** o **AF** (*Assured Forwarding*, RFC2597). Establece distintos niveles de prioridad organizados en 4 clases y 3 niveles por clase. Los nodos le asignan a cada clase unos recursos diferentes (de búfer y ancho de banda), consiguiendo con ello tratar a cada tráfico con distinta prioridad. Por otro lado, los paquetes de cada clase se organizan en 3 niveles de preferencia de descarte (*drop precedence*). Esto quiere decir que, en caso de congestión de los recursos asignados a alguna de las clases, el nodo descartará primero los paquetes que estén marcados con un nivel mayor. Por tanto, el comportamiento AF dispone de 12 niveles de prioridad. Los tres primeros bits del código definen la clase, los dos siguientes el nivel dentro de la clase, mientras que el último bit es siempre cero. El comportamiento AF no asegura caudales o retardos, aunque la probabilidad de pérdida de este tráfico es muy baja, aún en caso de congestión.
- **Por defecto** o **BE** (*Best Effort* o *Default Behaviour*). Es el nivel de prioridad más bajo, y se corresponde con el código 000000. Los nodos tratarán los paquetes que vengan marcados de esta forma como a cualquier otro paquete de datos. En caso necesario, también se han definido códigos DSCP para los servicios por defecto con prioridad (códigos 000xxx).

Adicionalmente, existe un conjunto de comportamientos conocidos como selector de clase (*Class Selector*). Estos se crearon para mantener la compatibilidad con la utilización actual del campo ToS de IPv4 y se corresponde con los códigos xxx000. Como en los casos anteriores, estos comportamientos asignan mayor prioridad a los códigos de mayor valor.

Aunque los códigos a aplicar a cada tipo de tráfico depende del administrador de cada dominio o de los acuerdos SLA que se tengan establecidos, para hacernos una idea, el tráfico de voz se podría marcar con la mayor prioridad EF (código DSCP 101110), el de vídeo con la prioridad AF41 (DSCP 100010), mientrus que al tráfico de señalización se le podrá asignar un prioridad baja (por ejemplo, 011000).

CÓDIGO	DESCRIPCIÓN
11xxx0	*Reservados (enrutado y control)*
101110	*Comportamiento EF (Expedited Forwarding)*
101xx0	*Configurable por el usuario*
100110	*Comportamiento AF clase 4 (AF43) con precedencia alta*
100100	*Comportamiento AF clase 4 (AF42) con precedencia media*
100010	*Comportamiento AF clase 4 (AF41) con precedencia baja*
100000	*Configurable por el usuario*
011110	*Comportamiento AF clase 3 (AF33) con precedencia alta*
011100	*Comportamiento AF clase 3 (AF32) con precedencia media*
011010	*Comportamiento AF clase 3 (AF31) con precedencia baja*
011000	*Configurable por el usuario*
010110	*Comportamiento AF clase 2 (AF23) con precedencia alta*
010100	*Comportamiento AF clase 2 (AF22) con precedencia media*
010010	*Comportamiento AF clase 2 (AF21) con precedencia baja*
010000	*Configurable por el usuario*
001110	*Comportamiento AF clase 1 (AF13) con precedencia alta*
001100	*Comportamiento AF clase 1 (AF12) con precedencia media*
001010	*Comportamiento AF clase 1 (AF11) con precedencia baja*
001000	*Configurable por el usuario*
000xx0	*Configurable por el usuario*
000000	*Por defecto (best effort)*
xxxx11	*Reservado para experimentación*
xxxx01	*Reservado*

Tabla 9.4. Códigos de los comportamientos del campo DS

RSVP. RESERVA DE RECURSOS

Aunque se han desarrollado distintas soluciones que utilizan la técnica IntServ de reserva de recursos, el protocolo más utilizado y que, de hecho, se ha convertido en el estándar para este tipo de servicio es RSVP. Este acrónimo, que significa *Resource ReserVation Protocol* o protocolo para la reserva de recursos, curiosamente, también coincide con el acrónimo francés RSVP que puede encontrarse a la entrada de muchos restaurantes y que significa *Réserve, S'il Vous Plaît*, 'Reserve, por favor'.

La técnica RSVP está descrita en la RFC2205 del IETF, publicado en septiembre de 1997. Mediante este protocolo es posible reservar los recursos necesarios de la red IP antes de que comience el intercambio de información multimedia entre los participantes. Para que esto sea posible, todos los nodos utilizados en la ruta, extremo a extremo, deben ser compatibles con RSVP.

El protocolo de señalización RSVP hace posible que se pueda ofrecer en redes IP (conmutación de paquetes) una calidad de servicio similar a la que se obtiene en las redes telefónicas tradicionales (conmutación de circuitos).

El protocolo RSVP basa su funcionamiento en el intercambio de mensajes entre el emisor (la parte que genera la información multimedia) y el receptor (la parte que recibe dicha información). Estos mensajes establecen una ruta de nodos intermedios entre uno y otro extremo de la comunicación que se conservan hasta el final de la misma.

La reserva de recursos comienza con el envío de un mensaje *Path* desde el terminal emisor hasta el receptor. Este mensaje va dejando una marca en los nodos por los que va pasando, dejando así establecida la ruta. Al llegar al destino, se genera una respuesta con un mensaje *Resv* que recorre en sentido inverso el mismo camino, reservando los recursos necesarios en cada nodo. Si se confirma la reserva de todos los tramos, se establece la comunicación, en caso contrario, el nodo afectado le enviará un mensaje de error al receptor.

RSVP realiza la reserva de recursos de forma unidireccional, lo que quiere decir que si el intercambio de información es bidireccional, se necesitará realizar una reserva de recursos para cada sentido.

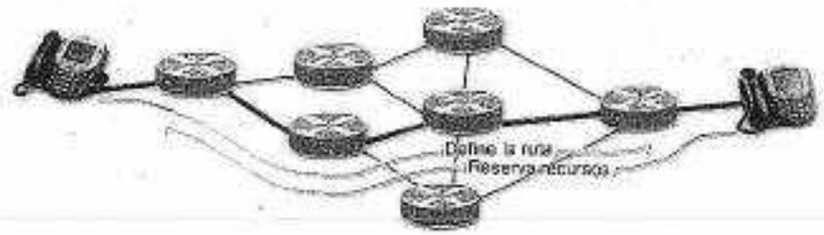

Figura 9.6. Proceso de reserva de recursos

El protocolo RSVP tiene en cuenta que una comunicación puede contar con más de un receptor (por ejemplo, en *multicast*). Como la información transmitida es la misma en los tramos de la ruta que comparten varios receptores, en estos tramos se realiza una única reserva de recursos. El hecho de que la reserva se realice desde el receptor hacia el emisor favorece esta labor.

Los nodos intermedios mantienen un registro de estado (*soft state*) de cada una de las comunicaciones, donde guardan la identificación de la sesión y los recursos reservados a la misma. Los registros de estado tienen un tiempo de expiración, lo que obliga a los emisores y receptores a enviar de forma periódica sus mensajes de mantenimiento de la reserva. Si un registro de estado cumple el tiempo de expiración, se elimina la reserva del nodo y se envía un mensaje de error a los participantes de la sesión.

Cada mensaje RSVP se compone de una serie de objetos, que contienen cada uno de ellos un tipo distinto de información. Por ejemplo, un objeto identifica la sesión, otro identifica al emisor, otro contiene los parámetros de calidad, etc. Dependiendo de la finalidad del mensaje se incluyen unos objetos u otros.

Los mensajes principales son *Path*, que es utilizado por el emisor para marcar la ruta, y *Resv*, utilizado por el receptor para solicitar la reserva de recursos. El resto de mensajes tienen que ver con la comunicación de errores y la eliminación de la reserva.

RFC	FECHA	DESCRIPCIÓN
RFC1633	06/1994	*Servicios integrados en la arquitectura Internet*
RFC2205	09/1997	*Especificación funcional de RSVP versión 1*
RFC2207	09/1997	*Extensiones RSVP para IPSec*
RFC2208	09/1997	*Despliegue de RSVP versión 1*
RFC2209	09/1997	*Reglas del procesamiento de mensajes de RSVP v1*
RFC2210	09/1997	*Uso de RSVP con los servicios integrados de IETF*
RFC2211	09/1997	*Servicio de carga controlada*
RFC2212	09/1997	*Servicio garantizado*
RFC2215	09/1997	*Parámetros de características de los servicios integrados*
RFC2748	01/2000	*Protocolo COPS*
RFC2749	01/2000	*Uso de COPS con RSVP*
RFC2750	01/2000	*Extensiones RSVP para el control de las políticas de uso*
RFC2998	11/2000	*Formato de los objetos RSVP*

Tabla 9.5. Ejemplos de RFC sobre el modelo de servicios integrados

Parámetros para la reserva de recursos

El protocolo RSVP tiene una forma particular de reservar recursos en un nodo, utiliza un sistema conocido como **cubo de testigos** (*token bucket*). Esto quiere decir que el nodo dispone de un registro (al que llama cubo) que almacena testigos. El cubo tiene una capacidad máxima de almacenamiento de testigos (b). Por otro lado, el cubo se va llenando automáticamente de testigos a una velocidad r (testigos por segundos).

Para regular el tráfico en una ruta determinada a su paso por el nodo, cada vez que se le da paso a un paquete se disminuye la capacidad del cubo en tantas unidades, tantos

testigos, como bytes tenga el paquete. Quiere esto decir que si se dispone de un tráfico de entrada P (bytes por segundos), menor que la velocidad r de recarga del cubo, dicho tráfico de entrada saldrá del *router* sin ningún retardo. El cubo se irá llenando poco a poco de testigos hasta alcanzar el valor b. Por el contrario, si P es mayor que r, el cubo se irá vaciando de testigos hasta que el número de testigos disponibles sea menor que el tamaño del paquete. En este caso, el paquete se desechará directamente o se pondrá en cola hasta que el cubo acumule el número de testigos suficientes.

Con este sistema, no sólo se garantiza un ancho de banda r, sino que se asumen ciertos picos a velocidades mayores durante un cierto tiempo. El tamaño y duración de los picos se regula por la capacidad del cubo (b). El resultado es que la velocidad media no será nunca superior a r, aunque puedan existir picos superiores que deben ser compensados con sus valles correspondientes.

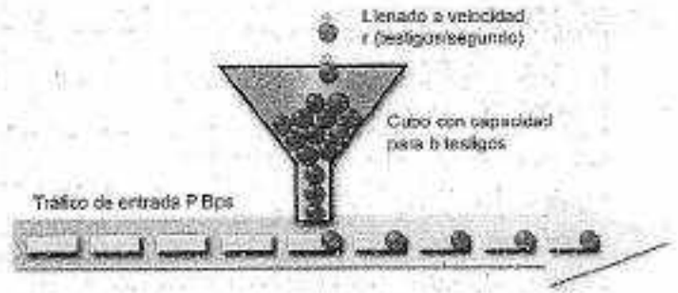

Figura 9.7. Símil del cubo de testigos

Aunque se utiliza el símil del cubo, en realidad, el sistema de cubo de testigos consiste en un registro numérico que incrementa su valor a la velocidad r hasta un valor máximo b. De la misma forma, se decrementa en una unidad por cada byte que deja pasar el *router*. Adicionalmente, los parámetros de definición de la calidad de servicio ofrecida con el sistema de cubo de testigos se complementan con la definición de los tamaños máximo (M) y mínimo (m) de los paquetes (en bytes), así como la velocidad máxima del tráfico de entrada (p). Por cierto, si algún paquete sobrepasa el tamaño M, el nodo no lo trata con la QoS solicitada, sino que lo enruta como el resto del tráfico de datos. Si, por el contrario, tuviera un tamaño menor que m, se le trataría como si el tamaño fuese m.

El conjunto de parámetros que definen las características del cubo de testigos se conoce como *TSpec* (*Traffic Specification*, 'Especificación del tráfico') y están recogidos en la RFC2215.

Otro parámetro que se utiliza en el control de la calidad de servicio es el conocido como *RSpec* (*Request Specification*, 'Especificación de la solicitud'), el cual está

formado por los valores R (*Rate*, 'Velocidad', en bytes por segundos) y S (*Slack*, 'Holgura', en microsegundos). El primero de ellos indica un ancho de banda solicitado. Este ancho de banda debe ser igual o mayor que r. Al ser mayor se está solicitando un ancho de banda extra, lo que mejora la calidad ofertada, disminuyendo las colas. El valor S indica la diferencia entre el retardo deseado y el retardo que se conseguiría si se utilizase el ancho de banda R.

Niveles de servicio

Utilizando los parámetros *TSpec* y *RSpec*, los nodos compatibles con RSVP pueden ofrecer dos niveles de servicio:

- **Carga controlada** (*Controlled load*). Utiliza mecanismos de control de admisión para garantizar recursos incluso cuando los nodos de red están sobrecargados. El resultado es un bajo retardo, aunque no existe una garantía total. El objetivo es que se ofrezca una calidad de servicio similar al que ofrecería la red cuando no está sobrecargada. Este tipo de calidad de servicio se conoce como *Soft QoS* y está recogido en RFC2211.
- **Garantizado** (*Guaranteed service*). En este caso, el protocolo garantiza la QoS en términos de ancho de banda, retardo extremo a extremo, fluctuaciones de retardo, etc. Este servicio está recogido en RFC2212 y requiere que todos los nodos de la red IP por los que pasa el flujo lo soporten.

Para implementar el servicio de carga controlada sólo es necesario definir los parámetros TSpec, mientras que para el servicio garantizado es necesario definir los parámetros TSpec y RSpec.

FORMATO DE LOS MENSAJES RSVP

Los mensajes RSVP utilizan el protocolo de transporte UDP sobre paquetes IP que tienen el número de protocolo IP 46. Al utilizar UDP no existe retransmisión de los paquetes perdidos o recibidos con error, no obstante, estos mensajes se retransmiten periódicamente, por lo que esto no suele ser un problema.

Los mensajes RSVP consisten en una cabecera común seguida por un cuerpo que está formado por un número variable de objetos. Los objetos contienen los distintos tipos de información que se desea intercambiar.

La cabecera común tiene una longitud fija de 8 bytes y está formada por los siguientes campos:

- ***Version*** (versión). Número de la versión de RSVP.
- ***Flags*** (indicador). No definido actualmente.
- ***Message Type*** (tipo de mensaje). Indica el tipo de mensaje RSVP. Actualmente están definidos los siguientes tipos:
  1 Path 2 Resv 3 PathErr 4 ResvErr ;
  5 PathTear 6 ResvTear 7 ResvConf.
- ***RSVP Checksum*** (suma de control RSVP). Suma de control del mensaje. Si no se utiliza suma de control se le da el valor 0.
- ***Send_TTL*** (TTL del envío). Es el valor TTL (*Time To Live*, 'Tiempo de vida') del mensaje. A este valor se le decrementa en una unidad en cada salto.
- ***RSVP_Length*** (longitud RSVP). Longitud total de este mensaje, incluyendo la cabecera común. La longitud de la cabecera es siempre 8 bytes.

Figura 9.8. Formato de los mensajes RSVP

En la cabecera de los paquetes IP se incluye el campo TTL que cada nodo va descontando en una unidad. Por otro lado, en la cabecera de los mensajes RSVP se incluye otro campo TTL que los nodos intermedios compatibles con RSVP también van descontando en una unidad. Si todos los nodos intermedios fueran compatibles con RSVP, el valor de ambos campos TTL debiera ser el mismo; en caso contrario, el TTL de la cabecera RSVP será mayor que el de la cabecera IP. Por tanto, el campo *Send_TTL* puede utilizarse para determinar si existe algún *router* no compatible con RSVP.

Tipos de mensajes

Aunque los mensajes más utilizados en RSVP son *Path* y *Resv*, el protocolo cuenta con los siguientes siete mensajes diferentes:

- ***Path*** (ruta). Utilizado por el emisor para iniciar el proceso de reserva de recursos. Con este mensaje se marca la ruta que se utilizará durante toda la sesión. El mensaje *Path* está formado por la cabecera común, seguido de varios objetos, entre los cuales, los obligatorios son *Session*, *Rsvp_Hop* y *Time_Value*, aunque también se pueden incluir estos otros: *Integrity*, *Policy_Data*, *Sender_Template*, *Sender_TSpec* o *ADSpec*.
- ***Resv*** (reserva). Lo utiliza el receptor para solicitarles a los nodos intermedios que reserven los recursos solicitados (los necesarios para obtener la calidad de servicio deseada). Este mensaje está formado por la cabecera común seguida por varios objetos, entre los que obligatoriamente deben estar *Session*, *Rsvp_Hop*, *Time_Value* y *Style*. Otros objetos que suelen incluirse en este mensaje son: *Integrity*, *Policy_Data*, *Scope*, *FlowSpec* y *Filter_Spec*.
- ***Path Teardown*** (clausura de la ruta). Se utiliza para eliminar los registros de ruta (*path state*) a lo largo de la ruta. Cada nodo elimina sus registros y le pasa el mensaje al nodo siguiente hasta llegar el receptor. Este mensaje lo puede enviar el emisor o cualquier nodo intermedio en el que haya expirado la reserva. Los objetos obligatorios para este mensaje son *Session* y *Rsvp_Hop*.
- ***Resv Teardown*** (clausura de la reserva). Utilizado para eliminar los registros de la reserva de los nodos (*reservation state*) a lo largo de la ruta. Cada nodo elimina sus registros y le pasa el mensaje al nodo siguiente hasta llegar al emisor. Este mensaje lo puede enviar el receptor o cualquier nodo intermedio en el que haya expirado la reserva. Los objetos obligatorios para este mensaje son *Session*, *Rsvp_Hop* y *Style*.
- ***Path Error*** (error de ruta). Se le envía al emisor para indicarle que ha habido un error con su mensaje *Path*. El mensaje *Path_Error* va pasando de nodo en nodo hasta llegar al emisor. La única finalidad de *Path_Error* es la de informar al emisor, por lo que no modifica ningún estado en los nodos de la ruta. Los objetos obligatorios que hay que incluir en este mensaje son *Session* y *Error_Spec*.
- ***Resv Error*** (error en la reserva). Se utiliza para notificarle al receptor de que ha habido un error con el mensaje *Resv*. El mensaje *Resv_Error* se va pasando de nodo en nodo hasta llegar al receptor. Los objetos obligatorios que se incluyen en este mensaje son *Session*, *Rsvp_Hop*, *Style* y *Error_Spec*.
- ***Confirmation*** (confirmación). Se utiliza para confirmarle al receptor que la reserva se ha realizado correctamente. Este mensaje incluye obligatoriamente los objetos *Session*, *Style*, *Resv_Confirm* y *Error_Spec*.

Como puede verse, además de los mensajes *Path* y *Resv* utilizados para reservar los recursos, existen mensajes para eliminar la reserva y para comunicar los posibles errores que hayan podido aparecer.

OBJETO	DEFINICIÓN	INFORMACIÓN INCLUIDA	Clase
INTEGRITY	Autentificación del nodo		4
SESSION	Identificación de la sesión	IP del destinatario, puerto del destinatario, identificación del protocolo (UDP)	1
RSVP_HOP	Identifica el salto siguiente/anterior	IP nodo RSVP siguiente/ anterior, interfaz lógico LIH	3
TIME_VALUE	Tiempo de refresco de	los registros de reserva (*soft states*)	5
POLICY_ DATA	Datos de política de uso		14
SENDER_ TEMPLATE	Identificación del emisor (en los mensajes PATH)	IP del emisor	11
SENDER_ TSPEC	Características del tráfico del emisor	Parámetros b, r, p, m y M	12
ADSPEC	Recoge los datos de la red (OPWA).	Latencia mínima, ancho de banda mínimo, número total de saltos, MTU, compatibilidad RSVP, parámetros del servicio garantizado o de carga controlada	13
SCOPE	Lista de hosts emisores		7
STYLE	Estilo de la reserva	Indicador, vector de opción (WF, FF o SE)	8
FLOWSPEC	Parámetros de la QoS deseada	Clase de servicio y parámetros TSpec (r, b, p, m, M) para carga controlada o TSpec y RSpec (R, S) para servicio garantizado.	9
FILTER_ SPEC	Identificación del emisor (en los mensajes RESV)	IP y puerto del emisor (o etiqueta de flujo, IPv6)	10
RESV_ CONFIRM	Identificación del solicitante de la QoS	IP del receptor	15
ERROR_ SPEC	Identificación del error	IP del nodo del error, código de error, indicador, valor de error	6

Tabla 9.6. Características de los objetos del protocolo RSVP

Formato de los objetos

Los objetos están formados por uno o más bloques de 32 bits donde el primero de ellos es la cabecera del objeto y el resto su contenido. La cabecera del objeto está formada por tres campos:

- ***Length*** (longitud). Longitud total del objeto en bytes.
- ***Class-Num*** (número de la clase). Se trata del número que identifica al objeto. Las clases de objetos son las siguientes:

1 Session	*3 Rsvp_Hop*	*4 Integrity*
5 Time_Value	*6 Error_Spec*	*7 Scope*
8 Style	*9 FlowSpec*	*10 Filter_Spec*
11 Sender_Template	*12 Sender_TSpec*	*13 Adspec*
14 Policy_Data	*15 Resv_Confirm*	

- *C_Type* (tipo de clase). Este número permite utilizar distintas versiones de un mismo objeto. Generalmente, identifican la versión del protocolo IP utilizado, por ejemplo:

1 IPv4	*2 IPv6 con número de puerto*	*3 IPv6 con etiqu. flujo*
4 IPv4/IPSec	*5 IPv6 con IPSec*	

Objetos de los mensajes RSVP

En la actualidad existen siete tipos distintos de mensajes RSVP. Cada uno de estos mensajes se compone de una serie de objetos obligatorios y otros optativos.

Los objetos más comunes utilizados por los mensajes RSVP son los siguientes:

- ***Integrity*** (Integridad). Contiene los datos criptográficos utilizados para autentificar el nodo que origina el mensaje, así como para verificar su contenido.
- ***Session*** (Sesión). Contiene la dirección IP del destinatario (*DestAddress*), la identificación del protocolo IP (46 para UDP) y el número de puerto del destinatario. Este objeto se utiliza para identificar la sesión a la que pertenece el mensaje y está presente en todos los mensajes RSVP.
- ***Rsvp_Hop*** (Salto RSVP). Contiene la dirección IP del nodo compatible RSVP que envió este mensaje, así como la identificación del manejador del interfaz lógico de salida por el que envió el mensaje. Este mensaje puede hacer referencia al nodo anterior de la ruta (*PHop* o *previous hop*, salto anterior), o al nodo siguiente de la ruta (*NHop* o *next hop*, salto siguiente), dependiendo de si el mensaje va del emisor al receptor o a la inversa. Se entiende que la ruta está definida desde el emisor al receptor. Esta información se guarda en cada nodo de la ruta para asegurarse de que todos los mensajes RSVP siguen la misma ruta que la que estableció el mensaje *Path*.

Longitud del objeto (16 bits)
Class-Num = 3
C-Type
Dirección IPv4/IPv6 del salto siguiente/anterior en la ruta
Manejador de interfaz lógica (LIH)

Figura 9.9. Formato del objeto Rsvp_Hop

- ***Time_Values*** (valores de tiempo). Contiene el valor del tiempo de refresco R utilizado por el creador del mensaje. Indica durante cuánto tiempo se considerará válido el mensaje (tiempo de expiración o *timeout*, en milisegundos). Se utiliza en todos los mensajes *Path* y *Resv*.

Figura 9.10. Formato del objeto Time_Values

- ***Policy_Data*** (datos de política de uso). Contiene información para que el módulo local de política de uso del nodo pueda decidir si la reserva asociada a este mensaje puede admitirse administrativamente. Este objeto puede aparecer en los mensajes *Path*, *Resv*, *PathErr* o *ResvErr*.
- ***Sender_Template*** (patrón del emisor). Se utiliza para identificar al emisor. Contiene la dirección IP del emisor y, opcionalmente, alguna información adicional de demultiplexación. Es obligatorio en los mensajes *Path*.
- ***Sender_TSpec*** (características del tráfico del emisor). Define las características del tráfico del flujo de datos del emisor. Es obligatorio en los mensajes *Path*. El tráfico se caracteriza por un algoritmo de cubo de testigos definidos por los parámetros b y r. El parámetro b es el tamaño del cubo y r la velocidad con la que se llena el cubo de testigos. Cuando se envía un paquete de m octetos se retiran m testigos del cubo. Además de los valores b y r, este objeto incluye también los valores p (velocidad máxima de los datos), m (tamaño mínimo del paquete) y M (tamaño máximo del paquete).

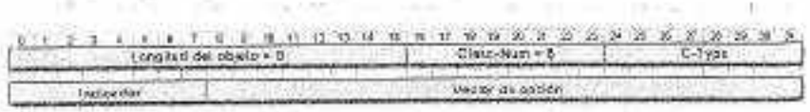

Figura 9.11. Formato del objeto Style

- ***AdSpec*** (información de la ruta). Este objeto se utiliza para dar a conocer al receptor las características de la ruta. Esto evita que el receptor solicite reservar unos recursos que la red no puede ofrecer. A este modelo se lo conoce como OPWA (*One-Pass With Advertisement*, una pasada con anuncio). Como este objeto pretende recoger los detalles de la red, sus valores se van actualizando conforme va pasando de nodo en nodo. Cuando llega al receptor le permite conocer las características de la ruta extremo a extremo y, por tanto, podrá determinar la calidad de servicio que puede solicitar razonablemente. Otro detalle interesante que incluye el objeto ADSpec es si algún nodo de la ruta no es compatible con RSVP.

Este objeto incluye tres fragmentos: el de los parámetros generales, el del servicio garantizado y el del servicio de carga controlada.

Los parámetros generales incluidos son: latencia mínima de la ruta (retardo), ancho de banda estimado de la ruta, número total de saltos, MTU (*maximum transmission unit*) de la ruta, y un bit para indicar si algún nodo de la ruta no soporta RSVP.

El fragmento del servicio garantizado contiene los parámetros que le permiten al receptor calcular el retardo extremo a extremo, un bit para indicar si algún nodo de la ruta no soporta el servicio garantizado, así como algún otro valor opcional.

El fragmento del servicio de carga controlada contiene un bit para indicar si algún nodo no es compatible con este servicio y algún valor opcional.

- ***Scope*** (alcance). Contiene una lista de los emisores a los que se envía el mensaje. Este objeto aparece en los mensajes *Resv*, *ResvErr*, o *ResvTear*.

Figura 9.12. Formato del objeto FlowSpec

- ***Style*** (estilo). Si la comunicación cuenta con distintos emisores o receptores, es posible definir distintas formas de hacer la reserva, distintos estilos. Este objeto define el estilo de la reserva (WF, FF o SE), así como alguna otra información adicional que no está presente en los objetos *FlowSpec* ni *Filter_Spec*. Este objeto es obligatorio en los mensajes *Resv*.
- ***FlowSpec*** (detalles del flujo). Se utiliza en los mensajes *Resv* para definir la calidad de servicio deseada. Para ello incluye un valor que indica la clase de servicio deseada (para el servicio garantizado toma el valor 2 y para el de carga controlada el valor 5, si el dato a enviar no está asociado con ningún servicio toma el valor 1) seguido por los parámetros necesarios para dicha clase. Para el servicio de carga controlada, los parámetros incluidos son lo de *TSpec* (r, b, p, m y M), mientras que para el servicio garantizado se incluyen los parámetros de *TSpec* y *RSpec* (R y S).
- ***Filter_Spec*** (detalles del filtro). Este objeto contiene la identificación del emisor (IP y puerto, o etiqueta de flujo en el caso de IPv6). Se utiliza en los mensajes *Resv* para identificar a los paquetes que forman parte del flujo de datos para el que se hace la reserva. Como el mensaje *Resv* también contiene la identificación del receptor (objeto *Session*), al añadir la identificación del emisor (el número IP) se está permitiendo que un mismo receptor pueda establecer más de una sesión simultánea con distintos emisores (por ejemplo, en una llamada a tres). El número de puerto del emisor es útil para establecer más de una sesión con un mismo emisor (por ejemplo, en el caso de una videoconferencia la información de audio irá a un puerto y la de vídeo a otro).

Longitud del objeto (16 bits)
Class-Num = 10
C-Type
Dirección IPv4/IPv6 del emisor
Número de puerto (16 bits) o Flow Label (24 bits)

Figura 9.13. Formato del objeto Filter_Spec

- ***Resv_Confirm*** (confirmación de receptor). Contiene la dirección IP del receptor que solicitó una confirmación. Este objeto aparece en los mensajes *Confirmation*.

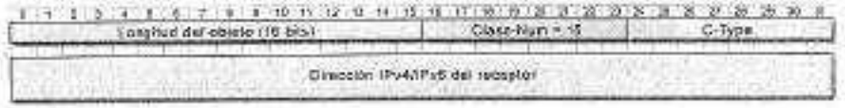

Figura 9.14. Formato del objeto Resv_Confirm

- *Error_Spec* (detalles del error). Los mensajes *PathErr* o *ResvErr* lo utilizan para identificar un error (incluye la dirección IP del nodo que detectó el error y el código de error), mientras que los mensajes *Confirmation* lo utilizan para especificar una confirmación.

Figura 9.15. Formato del objeto Error_Spec

Estilo de reserva

El objeto *Style* (estilo de reserva) se utiliza cuando participa más de un emisor en una sesión. En él se detalla el tipo de reserva que se desea realizar desde dos puntos de vista: el tratamiento que le dan los emisores a la reserva y cómo se van a identificar estos emisores.

En el primer caso, los emisores pueden hacer una única reserva compartida por todos los emisores o pueden realizar reservas independientes. En el segundo caso, todos los emisores pueden compartir una única identificación (conocida como comodín o *wildcard*) o cada emisor puede identificarse de forma independiente (utilizando una lista explícita).

		RESERVA (QoS deseada)	
		Por emisor	Compartida
IDENTIFICACIÓN DEL EMISOR	Independiente (Explícita)	Estilo FF (filtro independiente)	*Estilo SE (filtro explícito)*
	Compartida (Comodín)		*Estilo WF (filtro con comodín)*

Tabla 9.7. Estilos de reserva RSVP

La combinación de estas opciones da lugar a tres estilos de reserva:

- Filtro con comodín (WF, *Wildcard Filter*). En este caso, todos los emisores se identifican de forma compartida (comodín) y utilizan una reserva compartida.
- Filtro explícito compartido (SE, *Shared-Explicit Filter*). En esta modalidad cada emisor se identifica de forma independiente (explícita), pero utilizando una reserva compartida.
- Filtro independiente (FF, *Fixed-Filter*). Se utiliza una reserva independiente para cada emisor, identificándose, igualmente, cada uno de ellos de forma independiente.

Los dos primeros estilos de reserva son apropiados para las comunicaciones de voz, mientras que el último estilo, FF, es más apropiado para las comunicaciones de vídeo.

EL PROCESO DE LA RESERVA

Como se ha visto anteriormente, el proceso de reservar los recursos necesarios para el establecimiento de una comunicación multimedia comienza por definir la ruta de nodos entre el terminal emisor y receptor por los que se establecerá la comunicación. Una vez definida esta ruta, los paquetes RSVP (y, posteriormente, los de voz) se retransmitirán de nodo a nodo siguiendo siempre la misma ruta. Este circuito virtual queda definido mediante los registros correspondientes en cada nodo que forma la ruta.

En la idea de que los recursos no se queden reservados de forma innecesaria, estos registros tienen un tiempo de expiración que se va renovando cada vez que se recibe un mensaje *Path* o *Resv*. En el caso de que se dejen de recibir estos mensajes, se elimina el registro, aunque la comunicación no se haya dado por terminada con el mensaje correspondiente. Por tanto, el emisor y el receptor están continuamente enviando mensajes *Path* y *Resv* durante todo el tiempo que dura la comunicación.

Definir la ruta

El emisor envía un paquete *Path* al receptor (o receptores) de la comunicación. En este mensaje incluye los parámetros que identifican la sesión (objeto *Session*, que incluye la dirección del receptor), la identificación del emisor (objeto *Sender_Template*), las características del tráfico (objeto *Sender_TSpec*) y el valor del tiempo de refresco (*Time_Values*). El paquete va recorriendo la ruta hacia el receptor comprobando que todos los nodos intermedios (*routers*) son compatibles con RSVP (objeto *AdSpec*). Cada uno de estos nodos guarda un registro de la sesión (conocido como registro de estado de la ruta o *path state*) donde incluye: la identificación de la sesión, las características del tráfico y la dirección IP del nodo precedente. Cada nodo sustituye esta dirección IP (objeto *Resv_Hop*) por la suya propia antes de pasar el paquete al nodo siguiente.

El objeto *AdSpec* incluido en el mensaje *Path* va recogiendo distinta información de la ruta por la que va pasando (retardo, ancho de banda estimado, número de saltos, etc.) para que el receptor pueda conocer las características de la ruta.

Figura 9.16. El intercambio de mensajes RSVP

Reservar los recursos

Cuando el mensaje *Path* llega al receptor, éste tiene que decidir qué tipo de reserva va a solicitar. Para ello cuenta con las características del tráfico delimitadas por el emisor (objeto *Sender_TSpec*) e información sobre la ruta (objeto *AdSpec*). A continuación genera un mensaje *Resv* que envía de vuelta recorriendo la ruta anterior en sentido inverso. Este mensaje va reservando los recursos necesarios para la comunicación. Para ello, el mensaje *Resv* incluye la identificación de la sesión (objeto *Session*), el estilo de la reserva (objeto *Style*), los parámetros que definen la calidad de servicio deseada (objeto *FlowSpec*) y la identificación del emisor (objeto *Filter_Spec*). Esta información la van guardando los nodos intermedios en un registro conocido como estado de la reserva o *resv state*.

Figura 9.17. La reserva de recursos con RSVP

A los objetos *FlowSpec* y *Filter_Spec* se los conoce como descriptores de flujo (*Flow descriptor*). La forma cómo se incluyen estos objetos en el mensaje *Resv* depende del estilo de la reserva:

- WF (identificación y reserva conjunta). Como se realiza una única reserva y se cuenta con una única identificación de emisor, el descriptor de flujo será un objeto *FlowSpec*.
- FF (identificación y reserva independiente). Como se tiene una reserva independiente para cada emisor, se enviarán distintos mensajes de reserva incluyendo en cada uno de ellos un objeto *FlowSpec* seguido de *Filter_Spec*.
- SE (identificación independiente y reserva compartida). Se utiliza una única reserva, aunque se identifican los emisores de forma independiente. En este caso, el descriptor de flujo incluirá un objeto *FlowSpec* seguido de un objeto *Filter_Spec* por cada emisor.

Por otro lado, si la comunicación se realiza con más de un receptor, llega un momento en que un mismo nodo se encuentra con más de un mensaje *Resv* correspondiente a una misma sesión y a un mismo emisor. En este caso, el nodo junta todos estos mensajes de reserva y le asigna la solicitud de recursos más elevada.

Establecer la comunicación

Una vez que el mensaje Resv llega al emisor sin que haya habido ningún mensaje de error significará que han sido reservados todos los recursos necesarios a lo largo de la ruta, por lo que podrá comenzar el intercambio de información multimedia.

Hay que tener en cuenta que los paquetes RSVP incluyen las direcciones IP de origen y destino como cualquier otro paquete IP, por lo que si un nodo no es compatible con RSVP, simplemente lo retransmitirá como si se tratase de un paquete de datos.

En caso de error

Si alguno de los nodos no puede realizar la reserva solicitada, le envía al receptor un mensaje de error (mensaje *Resv_Error*). Igualmente, si el mensaje *Path* produjera algún error, el *router* correspondiente le enviaría un mensaje de error al emisor (mensaje *Path_Error*).

Es importante tener en cuenta que, en el caso de comunicaciones con más de un receptor, el fallo en la reserva de un receptor no debe influir en los otros. Por ejemplo, si llegan a un nodo dos peticiones de reserva realizadas por dos receptores, si el nodo rechaza una de ellas por solicitar un ancho de banda excesivo, la solicitud del otro receptor no tendría que fallar si su petición de ancho de banda es inferior y puede ser aceptada por el nodo.

Tanto el emisor como el receptor incluyen en los mensajes *Path* y *Resv* el objeto *Time_Values*. Este objeto contiene el periodo de refresco (R) para mantener la reserva activa. Cada nodo tiene un contador, y si no recibe un nuevo mensaje *Path* o *Resv* antes de que se cumpla este periodo de refresco eliminará la reserva. Por este motivo, tanto el emisor como el receptor envían mensaje *Path* y *Resv* de forma periódica.

Hay que tener en cuenta que, aunque el tiempo de refresco es R, los nodos tienen configurado un tiempo de espera mayor que R para contemplar la posibilidad de que se pierdan uno o más mensajes *Path* o *Resv* (dado que UDP no tiene garantía de entrega).

Terminar la comunicación

Cuando se da por terminada la comunicación, el emisor o el receptor envían un mensaje de cancelación de la reserva (*PathTear* y *ResvTear* respectivamente). Estos mensajes recorren toda la ruta cancelando los recursos reservados para dicho emisor o receptor. En el caso de contarse con más de un receptor, el mensaje *ResvTear* llegaría hasta el nodo de encuentro con otra reserva. Esto evita que desde un receptor se pueda cancelar la reserva de otro.

HERRAMIENTAS PARA LA CALIDAD DE SERVICIO

Como se ha visto, las técnicas de calidad de servicio se basan, fundamentalmente, en mejorar la eficiencia de los equipos que forman parte de la red. Para lograrlo, no sólo se disponen de técnicas específicas, como RSVP o la diferenciación de servicios, sino que se cuenta con todo un conjunto de herramientas encaminadas tanto a la clasificación,

tratamiento del flujo de tráfico y asignación de recursos, como a hacer que la red, en su conjunto, funcione eficazmente.

Las herramientas existentes se pueden clasificar en las siguientes categorías:

- **Clasificación de paquetes** (*packet classification*). Esta técnica consiste en clasificar los paquetes que van entrando en la red para tratarlos de forma distinta según su contenido. Esta clasificación se puede llevar a cabo analizando, tanto el campo DS, como otros campos de la cabecera IP (tipo de protocolo, direcciones IP y puertos), lo que permite asociar cada paquete a un tipo de tráfico distinto. Una vez hecha la clasificación es posible aplicar un tratamiento distinto en cuanto a priorización y asignación de recursos.
- **Utilización de colas** (*queue*). Consiste en crear distintas colas de tráfico, a las que se les asigna un nivel de prioridad distinto. La cola de alta prioridad podría enrutar el tráfico sin demora, mientras que la de baja prioridad lo haría cuando hubiera recursos disponibles. Es importante que el tráfico de una cola no bloquee completamente el tráfico de las otras. Por este motivo, existen distintos modelos de gestión de colas: desde el modelo simple de estricta priorización descrito anteriormente, hasta el que asigna distintos recursos a cada cola, pasando por otros más dinámicos. La asignación de los paquetes a cada tipo de cola se haría con técnicas de clasificación de paquetes.
- **Control de admisión** (*admission control*). Esta técnica consiste en admitir en cada clase sólo el tráfico al que se le puede garantizar la calidad de servicio prometida. El resto del tráfico se rechaza. Es algo así como si un avión sólo tiene ocho asientos en primera clase, si una novena persona solicita esta clase se le tiene que informar de que no hay plazas disponibles y que puede optar por una clase inferior o por viajar en otro momento.
- **Control de la política de uso** (*policy control*). A los elementos de red (tales como *routers* y *switches*) se les puede configurar para que sigan unas determinadas reglas basadas en criterios como la identidad del usuario, tipo de aplicación, ancho de banda, etc. Estas reglas pueden configurarse localmente en cada dispositivo o pueden residir en un servidor externo de política de uso. Estos servidores externos reciben el nombre de PDP (*Policy Decision Points*, 'Puntos de decisión de la política de uso'). Las reglas configuradas en los PDP suelen tener prioridad sobre las reglas configuradas localmente. A los elementos de red que aplican estas reglas se los conoce como PEP (*Policy Enforcement Points*, 'Puntos de aplicación de la política de uso'). El control de las políticas de utilización de los dispositivos de red son una parte fundamental de la calidad de servicio.
- **Aplicación de la política de uso** (*Policy Enforcement*). Los dispositivos de red deben contar con las herramientas necesarias que les permitan controlar, aplicar y hacer un seguimiento de la utilización de sus recursos con respecto a las políticas de uso que tengan configuradas.

Figura 9.18. Estructura interna de un nodo RSVP/COPS

Política de uso de los recursos

Una vez que se cuenta con mecanismos que permiten aplicar distinta QoS a distintos tipos de tráfico, el siguiente paso sería definir la calidad de servicio a aplicar a cada tipo de tráfico. De este asunto se ocupa lo que se ha venido a llamar política de calidad de servicio. Estas políticas especifican cómo se utilizan los mecanismos de diferenciación de tráfico y cómo se asignan los recursos.

Este punto es importante porque, no sólo se pueden aplicar distintos tratamientos a los distintos tipos de tráfico, sino que esta diferenciación también podría ser aplicable a distintos usuarios o a los distintos niveles de servicios contratados por los usuarios. Por ejemplo, podría haber usuarios de pago, que reciben una cierta QoS, mientras que el resto recibiría una QoS inferior. En ambos casos se trataría de transmisión de voz, pero el tratamiento sería distinto.

El IETF ha definido un protocolo conocido como COPS (*Common Open Policy Service*, 'Servicios comunes de políticas de uso abiertas'), especificado en la RFC2748, que se encarga de la administración, configuración y aplicación de las políticas de uso de la red.

COPS es un protocolo cliente-servidor que se compone de los siguientes elementos:

- PEP (*Policy Enforcement Point*, 'Punto de aplicación de la política de uso'). Se trata de los dispositivos que aplican las reglas (clientes COPS). COPS considera la posibilidad de definir distintos tipos de clientes.
- PDP (*Policy Decision Point*, 'Punto de decisión de la política de uso'). Se trata del servidor donde están configuradas las políticas de uso para todos los tipos de tráfico y de servicios (servidor COPS).

Cuando un PEP necesita aplicar cualquier regla se lo pregunta a un PDP para que tome la decisión oportuna. Mientras que el PEP viene a ser como la policía, el PDP hace el papel de juez.

Gracias a COPS se puede disponer de un modelo donde las políticas de uso se definen de forma centralizada. Los elementos PEP de la red, simplemente consultan los servidores PDP antes de tomar cualquier decisión. El protocolo COPS utiliza TCP como protocolo de transporte para asegurar la fiabilidad del intercambio de mensajes.

En el modelo COPS se contempla la posibilidad de que el cliente COPS disponga de una cierta autonomía de toma de decisión. La unidad local de toma de decisión se conoce como LPDP (*Local Policy Decision Point*, 'Punto de decisión local de la política de uso').

Control de la congestión

La congestión es la situación que se produce cuando se llenan las colas de un nodo, no pudiéndose almacenar los nuevos paquetes que llegan. En estas situaciones, los nuevos paquetes entrantes se desechan directamente, teniendo que ser las aplicaciones quienes detecten estas pérdidas y retransmitan los paquetes (si es el caso). El inconveniente de estas situaciones es tanto la pérdida de paquetes (UDP) como el retardo que supone su retransmisión (TCP), lo que conlleva una pérdida de calidad.

Para evitar en lo posible las situaciones de congestión se utilizan distintas técnicas de encolado y planificación. Esto supone disponer, en los nodos, de memorias intermedias (búfer) que guardan temporalmente los paquetes hasta que pueden ser retransmitidos, así como una lógica de priorización y retransmisión de los paquetes.

Entre las principales técnicas de encolado se encuentran las siguientes:

- **FIFO** (*First In, First Out*, 'El primero en entrar es el primero en salir'). En este caso, los paquetes se van retransmitiendo en el mismo orden en el que van llegando, sin distinciones. Por tanto, todos los paquetes que lleguen estando la cola saturada se desechan. Este es el esquema por defecto de casi todos los *routers*.

- **PQ** (*Priority Queuing*, 'Encolado con prioridad'). En este caso, los paquetes se marcan con distintas prioridades y se van almacenando en colas distintas, de forma que las colas con mayor prioridad se vacían más rápidamente que las que tienen una prioridad menor. El problema que tiene este sistema es que si existe mucho tráfico de alta prioridad nunca se transmite el tráfico de prioridad baja. A este fenómeno se le conoce como *traffic starvation* (hambre de tráfico).
- **CQ** (*Custom Queuing*, 'Encolado a medida'). En este caso también se crean colas con distintas prioridades, pero se define para cada una de ellas un número de paquetes mínimo para su retransmisión. De esta forma se garantiza un ancho de banda mínimo para cada cola.
- **WRR** (*Weighted Round Robin*, 'Asignación cíclica ponderada'). En este caso se le asigna un peso a cada cola y se reparte el ancho de banda disponible en función del peso de cada una. A este sistema también se lo conoce como CBQ (*Class Based Queuing*, 'Encolado basado en clases').

Pongámonos ahora en el caso de que le llega un paquete de 1500 bytes a un *router* y que se encuentra la cola vacía. El *router* comenzará su retransmisión sin demora, pero, en ese preciso momento, llega un paquete de alta prioridad. Como el paquete anterior ya empezó a ser retransmitido, hay que esperar a que termine, lo cual implica añadir un retardo considerable en la retransmisión del paquete de alta prioridad (si el enlace fuera de 128 Kbps, serían casi 100 milisegundos). Para evitar esta circunstancia se creó la técnica LFI (*Link Fragment and Interleaving*, 'Fragmentación e inserción'). Esta técnica consiste en trocear los paquetes en bloques de 10 milisegundos. De esta forma se limita el retardo máximo que puede sufrir un paquete en un nodo.

Claro que, si saber manejar la congestión cuando se produce es importante, también lo es poder evitarla (en lo posible). Pongámonos en otra circunstancia: cuando se congestiona un *router* se cortan muchas conexiones TCP de forma simultánea, lo que hace que se reduzca el caudal de forma abrupta. Esta reducción hace que desaparezca la congestión, pero, al comprobar las conexiones que hay caudal disponible, incrementan el tráfico, volviéndose a la situación de congestión. Para evitar este fenómeno oscilante existen mecanismos como RED (*Random Early Detection*, 'Detección temprana aleatoria'), WRED (*Weighted RED*, 'RED ponderada') o DWRED (*Distributed Weighted RED*, 'RED ponderada y distribuida') que evitan la congestión de la red descartando paquetes de forma inteligente antes de alcanzar la situación de congestión.

MPLS

Cuando un paquete llega a un nodo, lo primero que se hace es analizar su cabecera para determinar hacia dónde se tiene que enrutar, cuál será el siguiente salto. El nodo tiene que disponer de unos ciertos recursos para realizar este análisis (capacidad de proceso, memorias, etc.), que no son nada despreciables. Para resolver este problema, la

IETF publicó en enero de 2001 la recomendación RFC3031 que describía el protocolo conocido como MPLS (*Multiprotocol Label Switching*, 'Conmutación de etiquetas multiprotocolo').

MPLS le asigna una etiqueta a cada paquete que entra en la red. Esta etiqueta contiene toda la información necesaria para el encaminamiento. De esta forma, los nodos no tienen que leer toda la cabecera de cada uno de los paquetes, basta con que miren la etiqueta. La gran ventaja de MPLS es que consigue decisiones de enrutamiento más rápidas, lográndose, de esta forma, mejorar la QoS.

Los nodos que admiten el protocolo MPLS reciben el nombre de LSR (*Label Switched Router*, '*Router* de conmutación de etiquetas'). Al conjunto de LSR interconectados se lo conoce como dominio MPLS. Los nodos periféricos de este dominio se conocen como LER (*Label Edge Router*, '*Router* periférico de etiquetas'). Estos nodos son los que asignan las etiquetas a los paquetes entrantes en el dominio y las retiran en el momento de salir del mismo. A la hora de asignar la etiqueta se decide la ruta que va a seguir el paquete dentro del dominio MPLS. Para ello, se emplea la técnica de encaminamiento CBR (*Constrained Based Routing*, 'Enrutado basado en restricciones') que consiste en calcular las rutas óptimas en función de las condiciones del contorno.

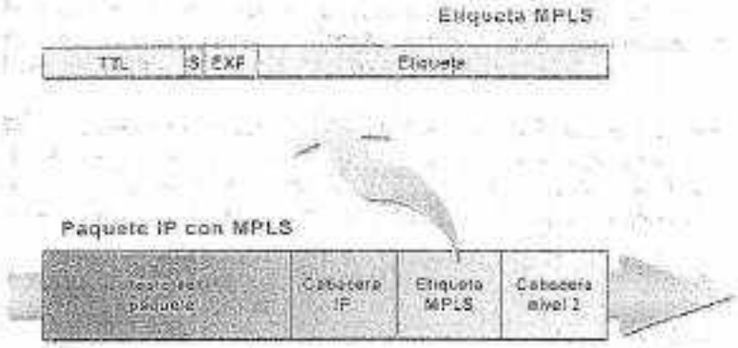

Figura 9.19. La etiqueta MPLS.

Las etiquetas MPLS se adjuntan en el paquete delante de la cabecera IP y lo identifican como perteneciente a una clase FEC (*Forwarding Equivalence Class*, 'Clase de paquetes equivalentes para el reenvío'). Todos los paquetes que pertenecen a una misma clase son tratados de la misma forma desde el punto de vista del enrutamiento. Por tanto, todos los paquetes de un mismo flujo de datos tendrán el mismo FEC y seguirán exactamente la misma ruta hasta llegar a su destino, llegando en el mismo orden

en que son transmitidos. Al camino que siguen los paquetes por el dominio MPLS se lo conoce como LSP (*Label Switched Path*, 'Ruta de los paquetes conmutados por etiquetas').

Con MPLS se puede asegurar también que a cada sesión se le asignarán los recursos que necesita. A esta técnica, con la que se puede asignar a un tipo específico de tráfico una ruta específica y los recursos de red necesarios se la conoce como ingeniería de tráfico (*traffic engineering*, RFC2702).

Diseño de soluciones de VoIP